NASA CONTRACTOR
REPORT

NASA CR-179264

A VELOCITY-PRESSURE INTEGRATED, MIXED INTERPOLATION, GALERKIN FINITE ELEMENT METHOD FOR HIGH REYNOLDS NUMBER LAMINAR FLOWS

By Sang-Wook Kim
Universities Space Research Association
Systems Dynamics Laboratory
Science and Engineering Directorate

Final Report

(NASA-CR-179264) A VELOCITY-PRESSURE INTEGRATED, MIXED INTERPOLATION, GALERKIN FINITE ELEMENT METHOD FOR HIGH REYNOLDS NUMBER LAMINAR FLOWS Final Report (Universities Space Research Association) 63/34 N88-18868 Unclas 0128799

February 1988

Prepared for
NASA-Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
CR-179264		

4. TITLE AND SUBTITLE	5. REPORT DATE
A Velocity-Pressure Integrated, Mixed Interpolation, Galerkin Finite Element Method for High Reynolds Number Laminar Flows	February 1988
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT #
Sang-Wook Kim*	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	11. CONTRACT OR GRANT NO. NAS8-35918
12. SPONSORING AGENCY NAME AND ADDRESS	13. TYPE OF REPORT & PERIOD COVERED Final Contractor Report
National Aeronautics and Space Administration Washington, D.C. 20546	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES Prepared for Systems Dynamics Laboratory, Science and Engineering Directorate
*Universities Space Research Association.

16. ABSTRACT

A velocity-pressure integrated, mixed interpolation, Galerkin finite element method for the Navier-Stokes equations is presented. In the method, the velocity variables have been interpolated using complete quadratic shape functions and the pressure has been interpolated using linear shape functions. For the two-dimensional case, the pressure is defined on a triangular element which is contained inside the complete bi-quadratic element for velocity variables; and for the three-dimensional case, the pressure is defined on a tetrahedral element which is again contained inside the complete tri-quadratic element. Thus the pressure is discontinuous across the element boundaries. Example problems considered include: a cavity flow for Reynolds number of 400 through 10,000; a laminar backward-facing step flow; and a laminar flow in a square duct of strong curvature. The computational results compared favorably with those of the finite difference methods as well as experimental data available. It was found that: the present method could capture the delicate pressure driven recirculation zones; the method yielded accurate velocity and pressure distributions; and the method required much fewer number of grid points than the finite difference methods to obtain comparable computational results. A finite element computer program (NSFLOW/L) for incompressible, laminar flows is also presented in this report.

ORIGINAL PAGE IS
OF POOR QUALITY

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Finite Element Method Navier-Stokes Equations Velocity-Pressure Integrated Method Finite Element Flow Analysis Program	Unclassified – Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	97	NTIS

For sale by National Technical Information Service, Springfield, Virginia 22161

ACKNOWLEDGMENTS

The author would like to express his thanks to Dr. C.-P. Chen of the University of Alabama in Huntsville for bringing the author's attention to many references for this work. Thanks are also due to Dr. Rand Decker for many suggestions in preparing the report.

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	FINITE ELEMENT EQUATIONS	2
	2.1 The Navier-Stokes Equations	2
	2.2 Method of Weighted Residuals and the Galerkin Finite Element Method	3
	2.3 Mixed Interpolation Methods for Velocity and Pressure	7
III.	EXAMPLE PROBLEMS	10
	3.1 Lid-Driven Cavity Flow	10
	3.2 Backward-Facing Step Flow	19
	3.3 Laminar Flow in a Square Duct of Strong Curvature	24
IV.	CONCLUSIONS AND DISCUSSION	31
REFERENCES		32
APPENDIX I.	Finite Element Computer Program (NSFLOW/L) for Incompressible, Laminar Flows	35
APPENDIX II.	Input Data for NSFLOW/L	73
	A.2.1 Cavity Flow for Re = 10,000	76
	A.2.2 Backward-Facing Step Flow	78
	A.2.3 Laminar Flow in a Square Duct of Strong Curvature	81
APPENDIX III.	Description of the Subroutines	91

LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Flow element	8
2.	Configuration, coordinates, and nomenclature of cavity flow	10
3.	Discretization of the cavity flow	12
4.	Velocity vectors for cavity flow	13
5.	Streamlines for cavity flow	14
6.	Pressure contours for cavity flow	15
7.	Horizontal velocity profiles for cavity flow at x = 0.5	16
8.	Configuration, coordinates, and nomenclature of backward-facing step flow	19
9.	Discretization of the backward-facing step flow	20
10.	Velocity vectors for the backward-facing step flow	21
11.	Streamlines for the backward-facing step flow	22
12.	Pressure contours for the backward-facing step flow	23
13.	Reattachment length versus Reynolds number	25
14.	Wall pressure for backward-facing step flow	26
15.	Configuration of the laminar flow in a square duct of strong curvature	27
16.	Discretization of the flow domain	28
17.	Velocity vectors on the curved section	29
18.	Secondary recirculation flows	30

LIST OF TABLES

Table	Title	Page
1.	Streamline Contour Label for Cavity Flow	17
2.	Pressure Contour Label for Cavity Flow	17
3.	Stream Function Values at the Center of Vortices for Cavity Flow	18
4.	Streamline Contour Label for Backward-Facing Step Flow.....	24
5.	Pressure Contour Label for Backward-Facing Step Flow	24

CONTRACTOR REPORT

A VELOCITY-PRESSURE INTEGRATED, MIXED INTERPOLATION, GALERKIN FINITE ELEMENT METHOD FOR HIGH REYNOLDS NUMBER LAMINAR FLOWS

I. INTRODUCTION

The finite element methods for the Navier-Stokes equations using the primitive variables of velocity and pressure can be categorized into three groups, in general. These are the velocity-pressure integrated mixed interpolation methods, the penalty methods, and the segregated velocity-pressure solution methods.

In the velocity--pressure integrated, mixed interpolation methods, the order of interpolating polynomial for velocity is chosen to be one order higher than that of pressure. Unfortunately, many velocity-pressure integrated, mixed interpolation methods yield inaccurate pressure which becomes more severe as the Reynolds number is increased. Details of the method can be found in Taylor and Hughes [1] and the references therein.

In the penalty method, the pressure is pre-eliminated from the Navier-Stokes equations by penalizing the conservation of mass equation, and hence, the continuity condition is approximately satisfied. If necessary, pressure can be recovered using the penalized conservation of mass equation in the post process. Details of various penalty methods and the computational results can be found in Zienkiewicz et al. [2], Engelman et al. [3], Kikuchi et al. [4], and Heinrich and Marshall [5], among many others.

Due to the shortcomings of these two classes of methods, and partly influenced by the success of the finite difference methods based on segregated formulation of the Navier-Stokes equations, such as the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithms [6], a few finite element methods adopting the segregated formulation of the Navier-Stokes equations began to appear in recent years [7,8,9]. But the computational results thus obtained did not show significant improvement in accuracy over the previous two classes of methods.

The finite element method has certain advantages over the finite difference methods. These advantages are: capability to model complicated domain more precisely, capability to include various boundary conditions more naturally, and capability to show convergence nature of the method mathematically. There exists a number of example cases for which the finite element method yielded more accurate computational results than the finite difference methods for low Reynolds number flows [10]. However, for high Reynolds number flows, especially when pressure driven recirculation zones exist in the flow field, the finite difference methods yielded more accurate computational results than the finite element methods. Thus, the advantages of the finite element method have not been well demonstrated for high Reynolds number viscous flows, as yet.

In this report, a velocity-pressure integrated, mixed interpolation, Galerkin finite element method for high Reynolds number flows is presented. The finite element system of equations for the Navier-Stokes equations has been obtained using the Galerkin finite element method, and the system of equations has been solved using a

frontal solver [1,11]. The present method yielded accurate computational results for low Reynolds number flows as well as high Reynolds number flows. It was also found that the method could capture subtle pressure driven recirculation zones, and that the computational results were free of numerical wiggles for high Reynolds number flows.

II. FINITE ELEMENT EQUATIONS

A finite element method for two- and three-dimensional steady, incompressible, laminar flows is described below. The method is based on the velocity-pressure integrated formulation of the Navier-Stokes equations using a mixed interpolation of velocity and pressure.

In the following discussions, consistent notations have been used throughout, and repeated indices imply summation over the indices, unless otherwise specified.

2.1 The Navier-Stokes Equations

The form of the Navier-Stokes equations used herein are given as:

$$\left.\begin{aligned} \rho u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} = b_i \qquad (1) \\ \frac{\partial u_j}{\partial x_j} = 0 \qquad (2) \end{aligned}\right\} \quad \text{in } \Omega$$

where

$$\tau_{ij} = 2\mu \epsilon_{ij} - p\delta_{ij} \quad , \qquad (3)$$

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad , \qquad (4)$$

Ω is the open bounded domain of the problem, the subscripts i and j denote the coordinate directions, ρ is the density, u_i is the velocity component in the i-th coordinate direction, p is the pressure, μ is the molecular viscosity of the fluid, b_i is the body force in the i-th coordinate direction, τ_{ij} is the stress tensor, ϵ_{ij} is the strain rate tensor, and δ_{ij} is the Kronecker delta such that $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$. The boundary conditions are given as:

$$\underset{\sim}{u} = \underset{\sim}{u}_o(\underset{\sim}{x}) \qquad \text{for } \underset{\sim}{x} \in \partial\Omega_1 \tag{5}$$

$$T_i = \tau_{ij} n_j \qquad \text{for } \underset{\sim}{x} \in \partial\Omega_2 \tag{6}$$

where $\underset{\sim}{x} = (x,y)$ for 2-D case and $\underset{\sim}{x} = (x,y,z)$ for 3-D case, $\partial\Omega_1$ is part of the boundary on which Dirichlet boundary condition is specified, $\partial\Omega_2$ is the rest of the boundary on which Neumann boundary condition is specified, and T_i is the surface traction.

2.2 Method of Weighted Residuals and Galerkin Finite Element Method

In the context of the method of weighted residuals, the test functions for the momentum equation and the conservation of mass equation are denoted as $W_u(\underset{\sim}{x})$ and $W_p(\underset{\sim}{x})$, respectively. The weak form of the Navier-Stokes equations are:

$$\int_\Omega W_u \left(\rho u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} - b_i \right) d\underset{\sim}{x} = 0 \tag{7}$$

$$\int_\Omega W_p \left(\frac{\partial u_j}{\partial x_j} \right) d\underset{\sim}{x} = 0 \tag{8}$$

Integrating by parts of the stress tensor term in equation (7) yields:

$$\int_\Omega \left(W_u \rho u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial W_u}{\partial x_j} \tau_{ij} - W_u b_i \right) d\underset{\sim}{x} - \int_{\partial\Omega_2} W_u \tau_{ij} n_j \, ds = 0 \tag{9}$$

where $\int_{\partial\Omega_1} W_u \tau_{ij} n_j ds$ term has been dropped out since the test function $W_u(\underset{\sim}{x})$ should vanish on the boundary $\partial\Omega_1$.

Introducing the finite element discretization into equations (9) and (8) yields:

$$\sum_{e=1}^{E} \int_{\Omega_e} \left[W_u \left(\rho u_j \frac{\partial u_i}{\partial x_j} \right) + \tau_{ij} \frac{\partial W_u}{\partial x_j} - W_u b_i \right] d\underset{\sim}{x} - \sum_{e=1}^{E} \int_{\partial\Omega_{e2}} W_u \tau_{ij} n_j ds = 0 \tag{10}$$

$$\sum_{e=1}^{E} \int_{\Omega_e} \left(W_p \frac{\partial u_j}{\partial x_j} \right) \, d\underset{\sim}{x} = 0 \tag{11}$$

where Ω_e is a finite element, E is the total number of elements, and $\partial\Omega_{e2}$ is the boundary of an element (Ω_e) which lie on the part of the boundary ($\partial\Omega_2$) for which the flux boundary condition has been specified.

The flux through inter-element boundary should be continuous and the normal vectors at the interface of the two adjacent elements are in the opposite directions. Therefore, all the inter-element fluxes in equation (10) cancel each other, and only the prescribed flux on the boundary $\partial\Omega_2$ contributes to the final system of equations. The consequence of eliminating the fluxes across inter-element boundaries in the finite element method is equivalent to enforcing the flux continuity condition across the inter-element boundaries.

Inserting equations (3), (4), and (6) into equation (10) yields:

$$\sum_{e=1}^{E} \int_{\Omega_e} \left[W_u \rho u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial W_u}{\partial x_j} \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{\partial W_u}{\partial x_j} p\delta_{ij} - W_u b_i \right] d\underset{\sim}{x}$$

$$- \sum_{e=1}^{E} \int_{\partial\Omega_{e2}} W_u T_i ds = 0 \quad . \tag{12}$$

The global finite element system of equations are obtained by assembling the element system of equations. Therefore, the discrete finite element system of equations are derived for an arbitrary element in the following discussions.

Let ϕ_m and ψ_n be sets of basic polynomials to interpolate velocities and pressure respectively, i.e.,

$$\left.\begin{aligned} u_i &= u_{im}\phi_m \quad , \qquad \text{sum on m, m = 1,M} \\ p &= p_n\psi_n \quad , \qquad \text{sum on n, n = 1,N} \end{aligned}\right\} \tag{13}$$

where u_{im} denotes the m-th nodal value of the velocity component u_i, p_n denotes the n-th nodal value of pressure, and M and N are the number of velocity nodes and the number of pressure nodes in an element, respectively.

In the Galerkin finite element method, the test functions are selected from the same space of interpolating polynomials as the trial functions. Then W_u and W_p can be expanded as:

$$\left.\begin{aligned} W_u &= a_k \phi_k \quad , \qquad \text{sum on } k,\ k = 1,M \\ W_p &= b_\ell \psi_\ell \quad , \qquad \text{sum on } \ell,\ \ell = 1,N \end{aligned}\right\} \tag{14}$$

where a_k and b_ℓ should vanish if a Dirichlet boundary condition has been specified for the corresponding degree of freedom; otherwise, a_k and b_ℓ are arbitrary constants.

The finite element system of equations for an element is obtained by substituting equations (13) and (14) into equations (12) and (11), and is given as:

$$\int_{\Omega_e} \left[\phi_k \rho u_{jq} \phi_q \frac{\partial \phi_m}{\partial x_j} u_{im} + \frac{\partial \phi_k}{\partial x_j} \mu \left(\frac{\partial \phi_m}{\partial x_j} u_{im} + \frac{\partial \phi_q}{\partial x_i} u_{jq} \right) - \frac{\partial \phi_k}{\partial x_j} \psi_n p_n \delta_{ij} - \phi_k b_i \right] d\underset{\sim}{x}$$

$$- \int_{\partial \Omega_{e2}} \phi_k T_i \, ds = 0 \tag{15}$$

$$\int_{\Omega_e} \psi_\ell \frac{\partial \phi_q}{\partial x_j} u_{jq} \, d\underset{\sim}{x} = 0 \tag{16}$$

where the subscript q ranges from 1 to M, and the arbitrariness of the coefficients a_k and b_ℓ have been made use of in deriving equations (15) and (16).

In matrix form, equations (15) and (16) are given as, for the three-dimensional case:

$$\left(\begin{bmatrix} \underset{\approx}{C} & \underset{\approx}{0} & \underset{\approx}{0} \\ (\text{sym}) & \underset{\approx}{C} & \underset{\approx}{0} \\ & & \underset{\approx}{C} \end{bmatrix} + \begin{bmatrix} \underset{\approx}{K}_0 & \underset{\approx}{0} & \underset{\approx}{0} \\ (\text{sym}) & \underset{\approx}{K}_0 & \underset{\approx}{0} \\ & & \underset{\approx}{K}_0 \end{bmatrix} + \begin{bmatrix} \underset{\approx}{K}_{11} & \underset{\approx}{K}_{12} & \underset{\approx}{K}_{13} \\ \underset{\approx}{K}_{21} & \underset{\approx}{K}_{22} & \underset{\approx}{K}_{23} \\ \underset{\approx}{K}_{31} & \underset{\approx}{K}_{32} & \underset{\approx}{K}_{33} \end{bmatrix} \right) \begin{Bmatrix} \underset{\sim}{u}_1 \\ \underset{\sim}{u}_2 \\ \underset{\sim}{u}_3 \end{Bmatrix}$$

$$- \begin{bmatrix} \underset{\approx}{Q}_1 \\ \underset{\approx}{Q}_2 \\ \underset{\approx}{Q}_3 \end{bmatrix} \{p\} = \begin{Bmatrix} \underset{\sim}{f}_1 \\ \underset{\sim}{f}_2 \\ \underset{\sim}{f}_3 \end{Bmatrix} + \{ \text{b.c.} \} \tag{17}$$

$$\begin{bmatrix} \underset{\sim}{Q}_1^T & \underset{\sim}{Q}_2^T & \underset{\sim}{Q}_3^T \end{bmatrix} \begin{Bmatrix} \underset{\sim}{u}_1 \\ \underset{\sim}{u}_2 \\ \underset{\sim}{u}_3 \end{Bmatrix} = \{ \underset{\sim}{0} \} \tag{18}$$

where

$$\underset{\approx}{C} = \int_{\Omega_e} \underset{\sim}{\phi}^T \rho \, (\underset{\sim}{u}_j^T \underset{\sim}{\phi}) \frac{\partial \underset{\sim}{\phi}}{\partial x_j} \, d\underset{\sim}{x} \quad , \tag{19}$$

$$\underset{\approx}{K}_0 = \int_{\Omega_e} \frac{\partial \underset{\sim}{\phi}^T}{\partial \underset{\sim}{x}_j} \mu \frac{\partial \underset{\sim}{\phi}}{\partial \underset{\sim}{x}_j} \, d\underset{\sim}{x} \quad , \tag{20}$$

$$\underset{\approx}{K}_{ij} = \int_{\Omega_e} \frac{\partial \underset{\sim}{\phi}^T}{\partial x_j} \mu \frac{\partial \underset{\sim}{\phi}}{\partial x_i} \, d\underset{\sim}{x} \quad , \tag{21}$$

$$\underset{\sim}{Q}_i = \int_{\Omega_e} \frac{\partial \underset{\sim}{\phi}^T}{\partial x_i} \underset{\sim}{\psi} \, d\underset{\sim}{x} \quad , \tag{22}$$

$$\underset{\sim}{f}_i = \int_{\Omega_e} \underset{\sim}{\phi}^T b_i \, d\underset{\sim}{x} \quad , \tag{23}$$

$\underset{\sim}{u}_i$ is a column vector of nodal values of the velocity component u_i, $\underset{\sim}{p}$ is a column vector of nodal pressure, $\underset{\sim}{\phi}$ is a column vector of interpolating polynomials for velocity, $\underset{\sim}{\psi}$ is a column vector of interpolating polynomials for pressure, {b.c.} is a column vector contributed by the specified flux boundary condition, and the subscripts i and j range from one to the number of spatial dimensions, respectively.

For the two-dimensional case, the element system of equations can be obtained by deleting the appropriate third row and column sub-matrices from equations (17) and (18).

The integrations in equations (19) through (23) were evaluated using the Gauss numerical quadrature method with three Gauss points for each coordinate direction. The assembled global system of equations was solved by a direct (Picard) iteration method using a frontal solver [1,12], and the solutions were updated using an under-relaxation method given as:

$$a_j^* = \alpha \, a_j^n + (1 - \alpha) \, a_j^{n-1} \tag{24}$$

where a_j represents any degree of freedom, α is the under-relaxation number, the superscripts n and n-1 denote iteration levels, and a_j^* is the updated solution. No under-relaxation was necessary for low Reynolds number flows. For high Reynolds number flows, $\alpha = 0.8$ and $\alpha = 1$ have been used for the velocities and the pressure, respectively.

2.3 Mixed Interpolation Methods for Velocity and Pressure

The pressure interpolation polynomials used in the present study are introduced in this section. For the two-dimensional case, the velocities are interpolated using the bi-quadratic shape functions and the pressure is interpolated using the linear shape functions defined on a triangular element which is contained inside the quadratic element, as shown in Figure 1(a). The three pressure nodes are located at the three Gauss points of the three-point Gauss quadrature rule for quadrilateral elements [13], i.e., the same locations as those used in the Reduced Integration Penalty (RIP) method. The coordinates of the pressure nodes on the computational element are given as:

$$\underset{\sim}{\xi}_n = \begin{cases} (\quad 0 \quad , \sqrt{2}/\sqrt{3}) & \text{for } n = 1 \\ (-1/\sqrt{2} , -1/\sqrt{6}) & \text{for } n = 2 \\ (\quad 1/\sqrt{2} , -1/\sqrt{6}) & \text{for } n = 3 \end{cases} \tag{25}$$

where $\underset{\sim}{\xi}_n = (\xi_n, \eta_n)$ for the two-dimensional case, and n denotes the pressure node number. The shape functions for each of the nodes are given as:

$$\left.\begin{aligned} \psi_1 &= \frac{1}{3} + \frac{\sqrt{2}}{\sqrt{3}}\,\eta \\ \psi_2 &= \frac{1}{3} - \frac{1}{\sqrt{2}}\,\xi - \frac{1}{\sqrt{6}}\,\eta \\ \psi_3 &= \frac{1}{3} + \frac{1}{\sqrt{2}}\,\xi - \frac{1}{\sqrt{6}}\,\eta \end{aligned}\right\} \tag{26}$$

For the three-dimensional case, the velocities are interpolated by the tri-quadratic shape functions and the pressure is interpolated using the linear shape functions defined on a tetrahedral element which is contained inside the tri-quadratic brick element, as shown in Figure 1(b). The coordinates of the pressure nodes on the computational element are given as:

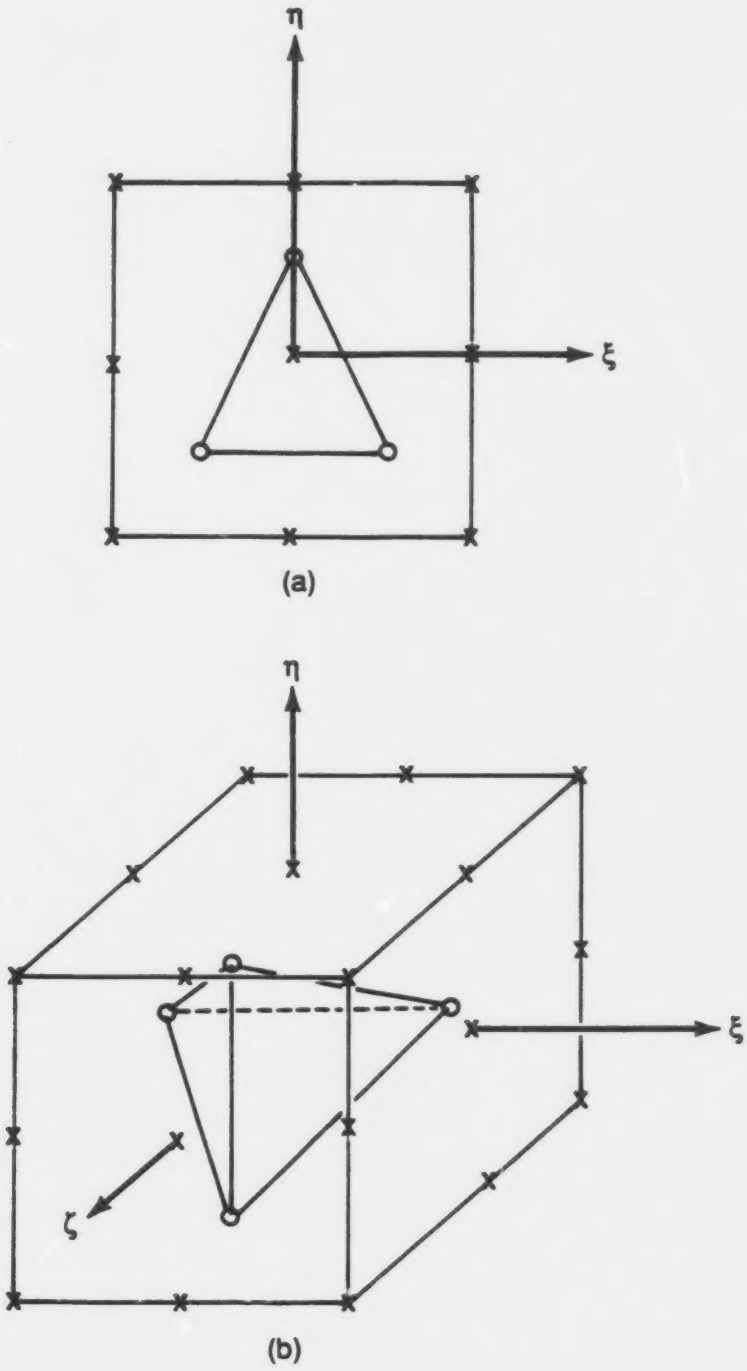


Figure 1. Flow elements. (a) Two-dimensional element, (b) three-dimensional element, x: velocity nodes, o: pressure nodes.

$$\xi_n = \begin{cases} (\ 0\ ,\ \sqrt{2}/\sqrt{3}\ ,\ -1/\sqrt{3}\ ) & \text{for } n = 1 \\ (\ 0\ ,\ -\sqrt{2}/\sqrt{3}\ ,\ -1/\sqrt{3}\ ) & \text{for } n = 2 \\ (\ \sqrt{2}/\sqrt{3}\ ,\ 0\ ,\ 1/\sqrt{3}\ ) & \text{for } n = 3 \\ (-\sqrt{2}/\sqrt{3}\ ,\ 0\ ,\ 1/\sqrt{3}\ ) & \text{for } n = 4 \end{cases} \tag{27}$$

where $\xi_n = (\xi_n, \eta_n, \zeta_n)$ for the three-dimensional case, and n denotes the node numbers of the pressure nodes. The shape functions for each of the pressure nodes are given as:

$$\left.\begin{aligned} \psi_1 &= \frac{1}{4} + \frac{1}{2}\frac{\sqrt{3}}{\sqrt{2}}\,\eta - \frac{\sqrt{3}}{4}\,\zeta \\ \psi_2 &= \frac{1}{4} - \frac{1}{2}\frac{\sqrt{3}}{\sqrt{2}}\,\eta - \frac{\sqrt{3}}{4}\,\zeta \\ \psi_3 &= \frac{1}{4} + \frac{1}{2}\frac{\sqrt{3}}{\sqrt{2}}\,\xi + \frac{\sqrt{3}}{4}\,\zeta \\ \psi_4 &= \frac{1}{4} - \frac{1}{2}\frac{\sqrt{3}}{\sqrt{2}}\,\xi + \frac{\sqrt{3}}{4}\,\zeta \end{aligned}\right\} \tag{28}$$

The shape functions given in equations (26) and (30) satisfy the relationship:

$$\psi_\ell(\xi_n) = \begin{cases} 1 \text{ if } \ell = n \\ 0 \text{ if } \ell \neq n \end{cases} \tag{29}$$

where ψ_ℓ is the shape function for the ℓ-the pressure node and ξ_n denotes the coordinates of the n-th pressure node. An additional pressure interpolation method tested for two-dimensional flows was $\underset{\sim}{\psi}^T = \{1, x, y\}$. This pressure interpolation method yielded the same computational results, up to several significant digits, as equation (26). However, the pressure interpolation polynomials given in equation (26) exhibited better convergence behavior than the other method as the number of iterations were increased, see Reference 14.

III. EXAMPLE PROBLEMS

The finite element method described in the previous sections has been tested and validated by solving a few example problems for which a vast amount of computational results and/or experiment data were available. These are: a lid driven cavity flow [5,15-19], a laminar backward-facing step flow [20-22], and a laminar flow in a square duct of strong curvature [24,25].

In the following discussions, solving the coupled momentum equation and the conservation of mass equation once is counted as an iteration. The convergence criterion used was

$$\left| 1 - \frac{a_{i,j}^{n}}{A_i^{n-1}} \right| < \epsilon \quad , \quad i = \{u, v, w, \text{ or } p\} \text{ and } j = 1,T \tag{30}$$

where $a_{i,j}$ denotes the i-th flow variable for the j-th node; A_i^{n-1} is the maximum nodal value of the i-th flow variable in the previous iteration; T is the total number of nodes; and ϵ is the convergence criterion. $\epsilon = 1 \times 10^{-4}$ and $\epsilon = 1 \times 10^{-3}$ have been used for the two- and three-dimensional flows, respectively.

For the mixed interpolation methods used herein, the pressure is discontinuous across element boundaries. Thus the nodal pressure at the velocity node has been obtained by averaging all the pressure contributions made by the elements containing the node; and each of the contributions has been computed using equation (13).

3.1 Lid Driven Cavity Flow

The cavity flow is described in Figure 2. The no slip boundary condition, i.e., $u = v = 0$, has been applied at all the boundaries except at $y = 1$ where $u = 1$ and $v = 0$. A Dirichlet pressure boundary condition has been specified at an arbitrary

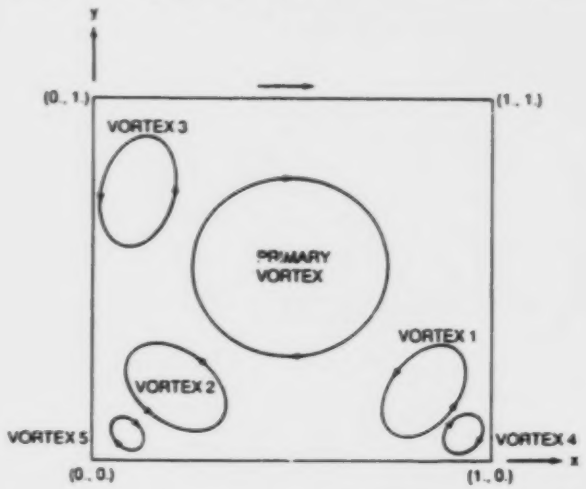


Figure 2. Configuration, coordinates, and nomenclature of cavity flow.

pressure node in the flow domain. The Reynolds numbers considered were 400, 1000, 3200, 5000, 7500, and 10,000; where the Reynolds number is defined as $R_e = \rho UL/\mu$, $U = 1$ is the velocity of the lid, and $L = 1$ is the reference length. The various Reynolds numbers have been obtained by varying the molecular viscosity (μ) of the fluid, with the rest of the parameters kept as constants. Two different grids, as shown in Figure 3, were used.

Computation of the cavity flow was carried out in the order of increasing the Reynolds number. Uniform zero values for both velocity and pressure were used as an initial guess for Re = 400, the converged solution for Re = 400 was used as an initial guess for Re = 1000, and so on. The required number of iterations were 17, 19, 21, 21, 26, and 30, for Reynolds numbers of 400, 1000, 3200, 5000, 7500, and 10,000, respectively.

For the coarse grid case (Grid A), convergent solutions were obtained up to Reynolds number of 3200. But the coarse grid could not resolve the smallest eddies (vortices 4 and 5 in Figure 2), and thus the grid was not tested for Re > 3200. Velocity vectors obtained by using the coarse grid for $Re \leq 1000$ are shown in Figures 4(a) and 4(b), respectively. The streamline contours for $Re \leq 1000$ obtained by using the coarse grid were qualitatively the same as those obtained by using the fine grid (Grid B), but the magnitude of the minimum stream function values at the center of the primary vortex was a few percent smaller than those obtained by using the fine grid.

The velocity vectors ($Re \geq 3200$) and the streamline contours obtained by using the fine grid are shown in Figures 4 and 5, respectively. The streamline contour labels are given in Table 1. In Figure 5, it can be seen that the sizes of vortices 4 and 5 of the present computational results are smaller than those of Schreiber and Keller [16] and Ghia et al. [17].

The normalized pressure contours for all the Reynolds number cases obtained by using the fine grid are shown in Figure 6, and the pressure contour labels are given in Table 2. The normalized pressure (P) has been obtained from the static pressure (p) using a relationship given as $P = pL_{ref}/V_{ref}/\mu$, where $L_{ref} = 1$ is the reference length, $V_{ref} = 1$ is the reference velocity, and μ is the molecular viscosity of the fluid.

The horizontal velocity profiles at $x = 0.5$ are compared with various computational results in Figure 7. For Re = 400, the magnitude of the minimum horizontal velocity due to Burggraf [15] was approximately 12 percent smaller than that of Ghia et al. [17], and the same velocity due to Bercovier and Engelman [19] was approximately 25 percent smaller than that of Ghia et al. [17]. For Re = 1000, the horizontal velocity profile due to Bercovier and Engelman [19] further deviated from that of Ghia et al. or the present computational results. In general, the present computational results compared more favorably with those of Ghia et al. than those of Burggraf [15] and Bercovier and Engelman [19], as shown in Figure 7.

For Re = 10,000, it can be seen that the horizontal velocity profile due to Schreiber and Keller [16] is significantly different from that of Ghia et al. [17], which shows that the global computational results can be significantly different depending on the order of difference approximation used for the boundary condition. The present computational results compared more favorably with those of Ghia et al. [17] than those of Schreiber and Keller [16].

The local maximum and minimum values of the stream function at the center of the primary vortex and the first three secondary vortices are compared with those of References 16 to 18 in Table 3. It can be seen that the present computational results, obtained by using approximately one-fourth of the number of grid points used in References 16 and 17, compared favorably with these data, in general.

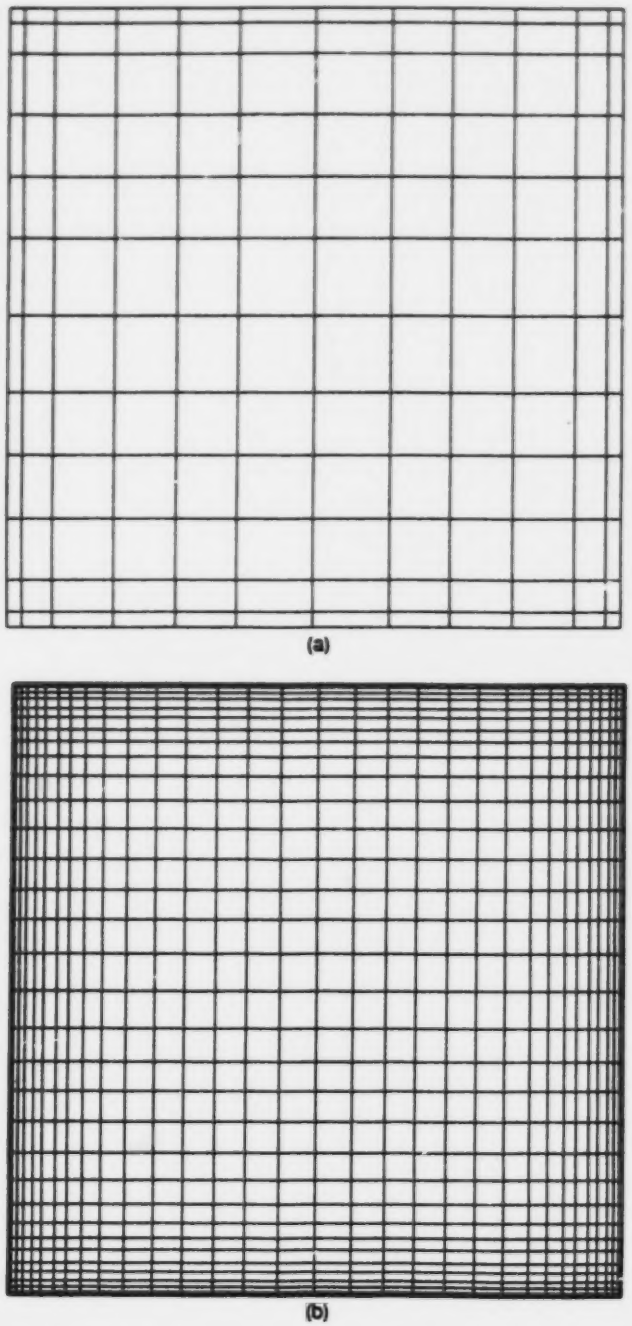
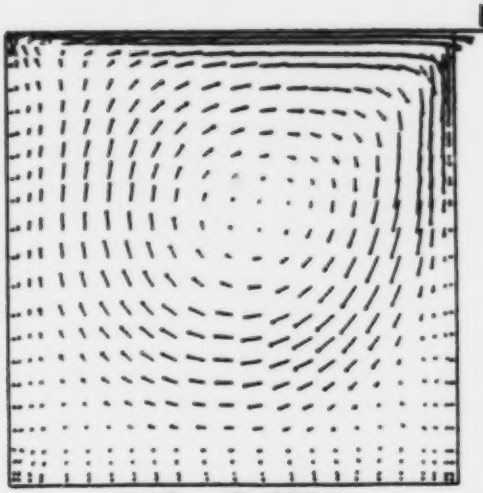
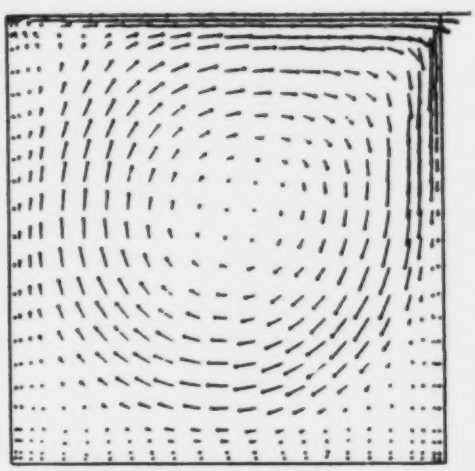


Figure 3. Discretization of cavity flow.
(a) Grid A, 25x25 grids (12x12 quadratic elements).
(b) Grid B, 65x65 grids (32x32 quadratic elements).

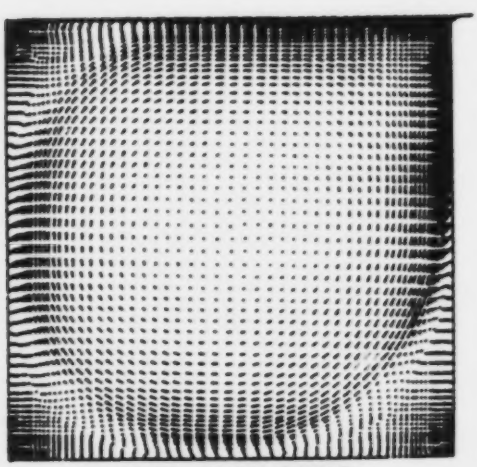
ORIGINAL PAGE IS
OF POOR QUALITY



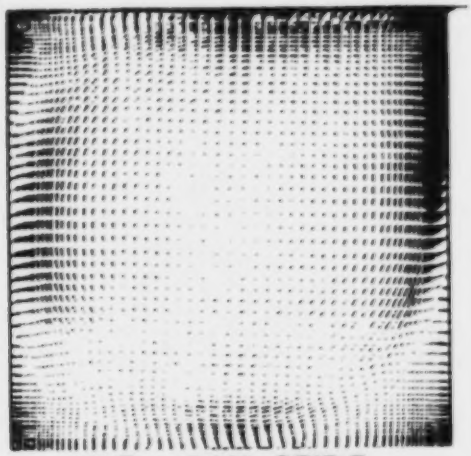
(a) Re = 400, GRID A



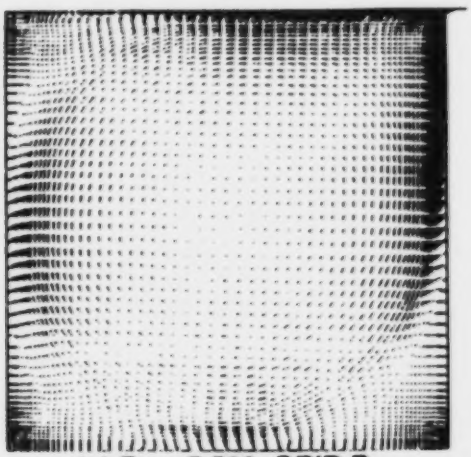
(b) Re = 1,000, GRID A



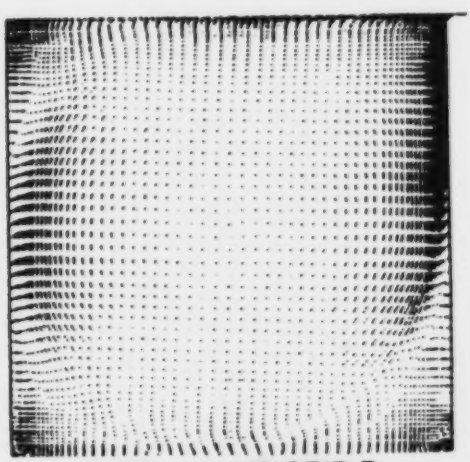
(c) Re = 3,200, GRID B



(d) Re = 5,000, GRID B



(e) Re = 7,500, GRID B



(f) RE = 10,000 GRID B

Figure 4. Velocity vectors for cavity flow, Re: Reynolds number.

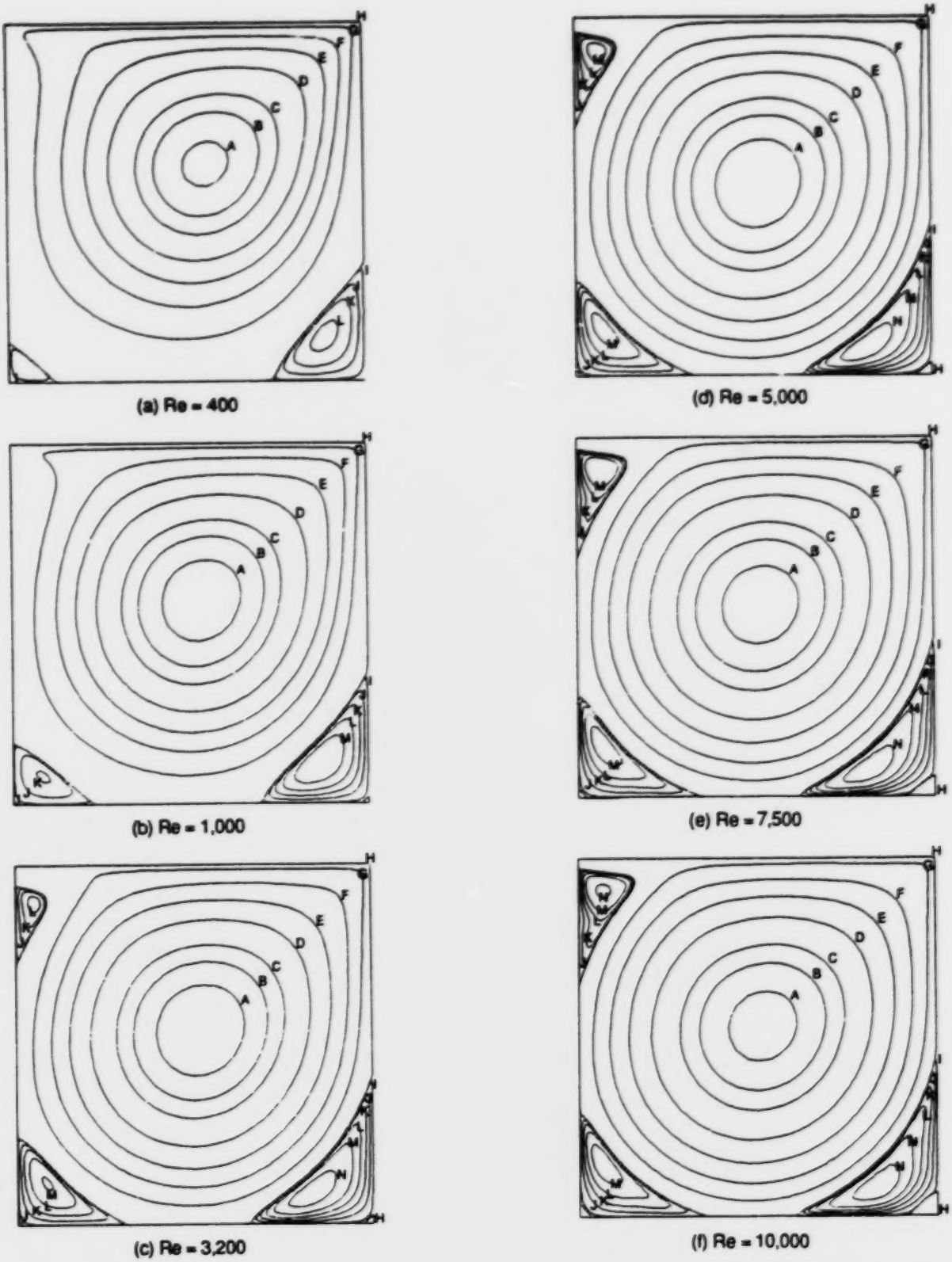


Figure 5. Streamlines for cavity flow.

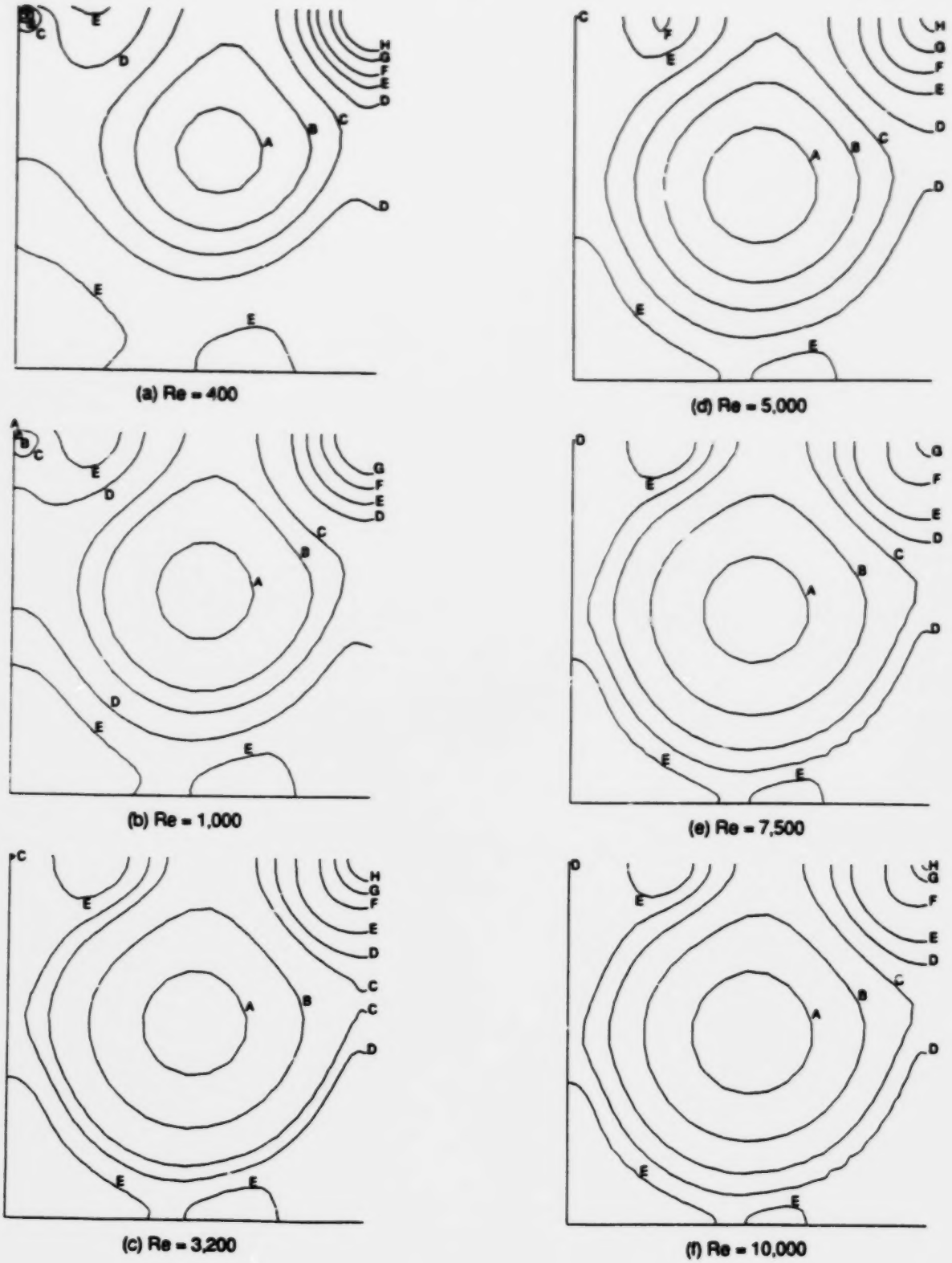


Figure 6. Pressure contours for cavity flow.

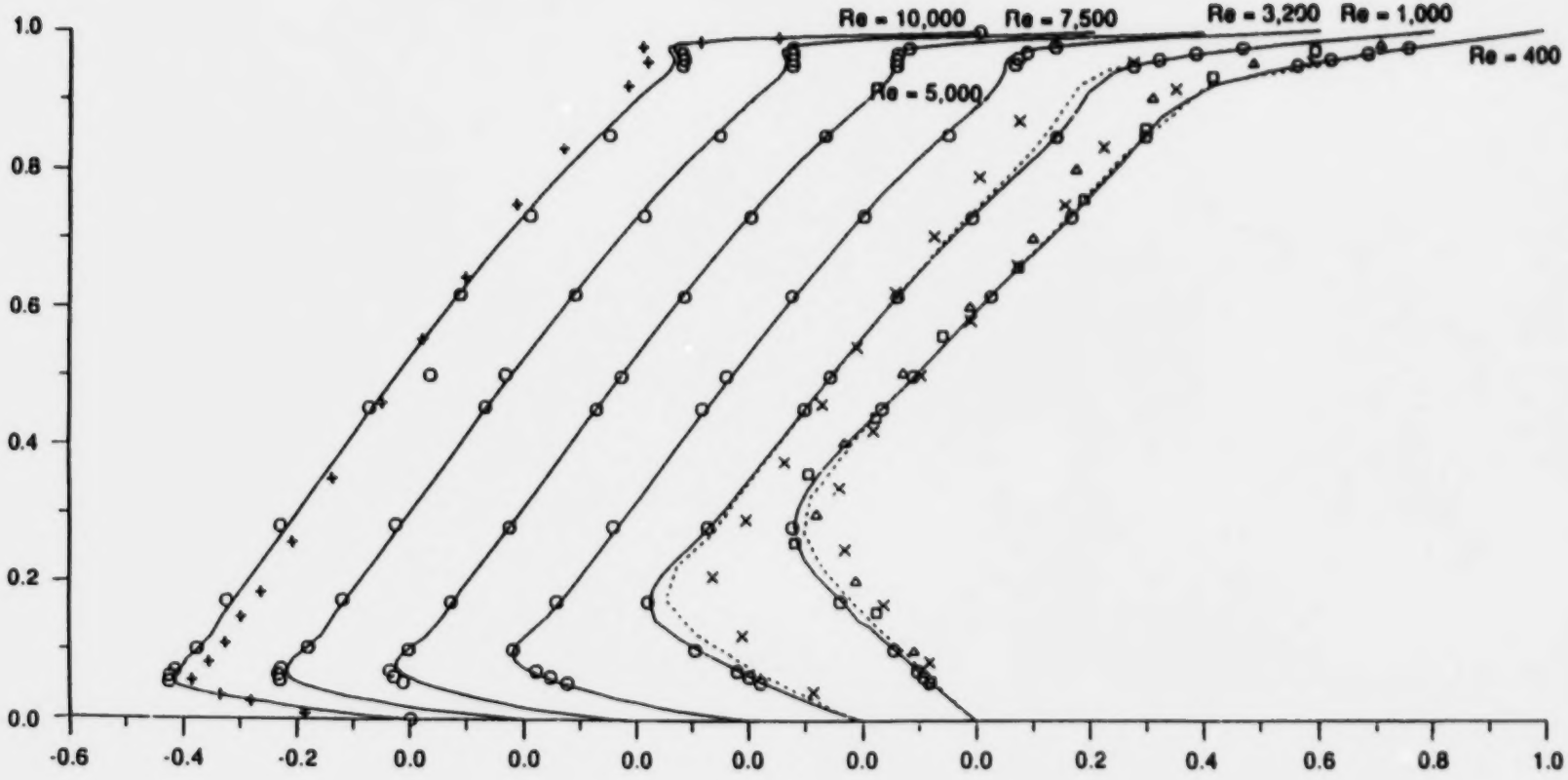


Figure 7. Horizontal velocity profiles of cavity flow at x = 0.5; ----: present (coarse grid), ——: present (fine grid, O: Ghia et al. (129x129 grids), x: Bercovier and Engelman (25x25 grids), Δ: Burggraf (41x41 grids), :Heinrich and Marshall (41x41 grids), +: Schreiber and Keller (180x180 grids).

TABLE 1. STREAMLINE CONTOUR LABEL FOR CAVITY FLOW

Label	ψ^*	Label	ψ	Label	ψ
A	-0.11	F	-0.03	K	$2.X10^{-4}$
B	-0.10	G	-0.01	L	$5.X10^{-4}$
C	-0.09	H	$-1.X10^{-10}$	M	$1.X10^{-3}$
D	-0.07	I	$1.X10^{-6}$	N	$2.X10^{-3}$
E	-0.05	J	$5.X10^{-5}$		

* ψ: stream function

TABLE 2. PRESSURE CONTOUR LABEL FOR CAVITY FLOW

Label	Reynolds Number (Re)					
	400	1000	3200	5000	7500	10000
A	-40.	-100.	-310.	-470.	-700.	-900.
B	-30.	-70.	-220.	-370.	-500.	-650.
C	-20.	-50.	-120.	-270.	-300.	-400.
D	-10.	-20.	-70.	-150.	-150.	-200.
E	0.	0.	0.	0.	0.	0.
F	10.	30.	100.	120.	300.	400.
G	30.	80.	200.	280.	1000.	1000.
H	50.	---	400.	900.	---	3000.

TABLE 3. STREAM FUNCTION VALUES AT THE CENTER OF VORTICES FOR CAVITY FLOW

	Re	Schreiber* & Keller	Ghia° et. al.	Gresho× et. al.	present+
Primary Vortex	400	-0.1130	-0.1139	---	-0.1128
	1000	-0.1160	-0.1179	-0.114	-0.1169
	3200	---	-0.1204	-0.118	-0.1181
	5000	---	-0.1190	-0.109	-0.1173
	7500	---	-0.1200	-0.108	-0.1157
	10000	-0.1028	-0.1197	-0.101	-0.1150
Vortex 1	400	6.440×10^{-4}	6.4235×10^{-4}	---	6.1810×10^{-4}
	1000	1.700×10^{-3}	1.7510×10^{-3}	1.760×10^{-3}	1.6594×10^{-3}
	3200	---	3.1396×10^{-3}	3.290×10^{-3}	2.6744×10^{-3}
	5000	---	3.0836×10^{-3}	3.870×10^{-3}	2.7786×10^{-3}
	7500	---	3.2848×10^{-3}	4.860×10^{-3}	2.7396×10^{-3}
	10000	2.960×10^{-3}	3.4183×10^{-3}	5.540×10^{-3}	2.7528×10^{-3}
Vortex 2	400	1.450×10^{-5}	1.4195×10^{-5}	---	1.3577×10^{-5}
	1000	2.170×10^{-4}	2.3113×10^{-4}	2.0×10^{-4}	2.1951×10^{-4}
	3200	---	9.7820×10^{-4}	1.20×10^{-3}	1.0465×10^{-3}
	5000	---	1.3612×10^{-3}	1.490×10^{-3}	1.2675×10^{-3}
	7500	---	1.4671×10^{-3}	1.750×10^{-3}	1.3697×10^{-3}
	10000	---	1.5183×10^{-3}	1.930×10^{-3}	1.4055×10^{-3}
Vortex 3	3200	---	7.2786×10^{-4}	5.860×10^{-3}	6.4440×10^{-4}
	5000	---	1.4564×10^{-3}	1.230×10^{-3}	1.3045×10^{-3}
	7500	---	2.0462×10^{-3}	1.840×10^{-3}	1.8426×10^{-3}
	10000	---	2.4210×10^{-3}	2.230×10^{-3}	2.1817×10^{-3}

* 141x141 grids for Re = 400 and 1000, and 180x180 grids for Re=10,000.
o 257x257 grids for Re = 400, 5000, 7500, and 10,000, and 129x129 grids for Re = 1000 and 3200.
x 51x51 grids for all Reynolds numbers.
+ 65x65 grids for all Reynolds numbers.

3.2 Backward-Facing Step Flow

A laminar backward-facing step flow with an expansion ratio of 1:1.94 is considered below. A description of the problem is shown in Figure 8, and the experimental data can be found in Armaly et al. [20]. The Reynolds number, $Re = \rho VD/\mu$, was based on the hydraulic diameter (D = 0.0104 m) and the bulk velocity (V = 0.6667 m/sec) at the inlet. The various Reynolds numbers have been obtained by varying the molecular viscosity (μ) of the fluid, with the rest of the parameters kept as constants. The velocity profile of a fully developed channel flow has been prescribed at the inlet; and the vanishing normal stress has been applied at the exit boundary.

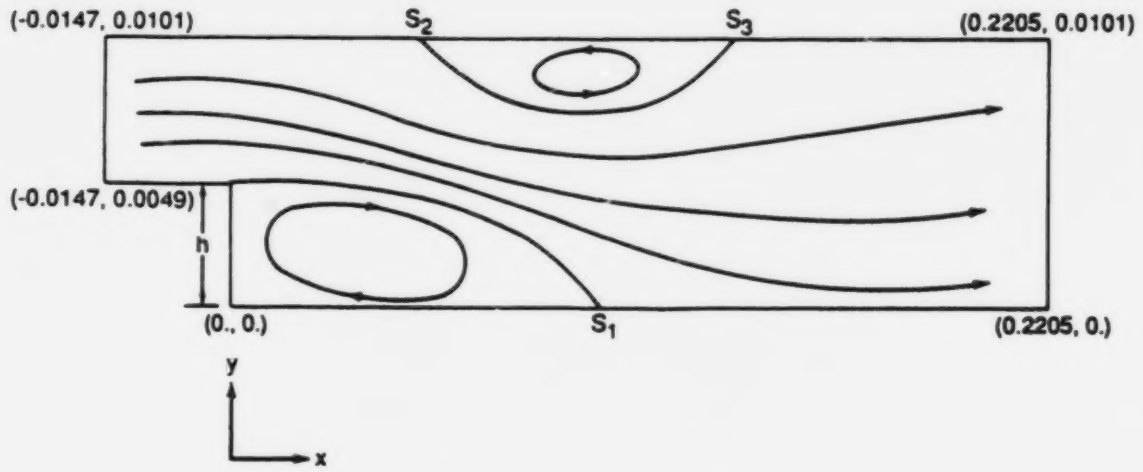
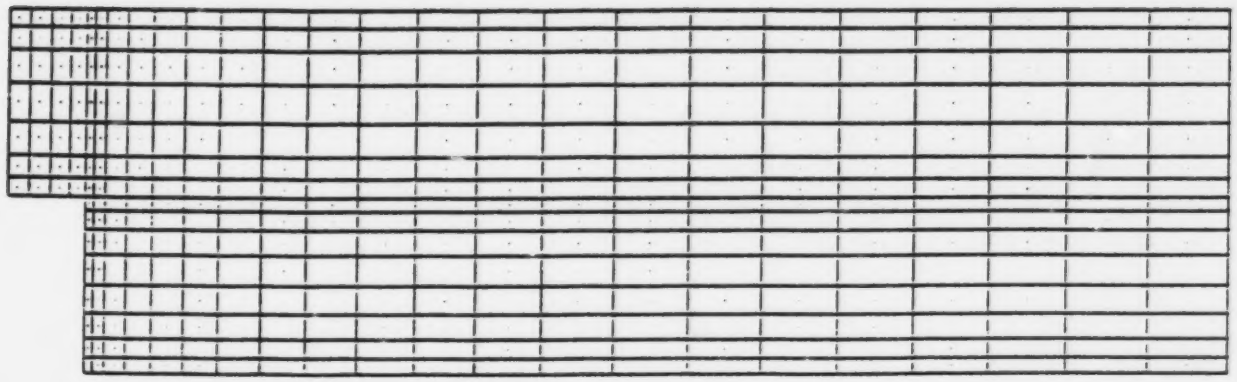


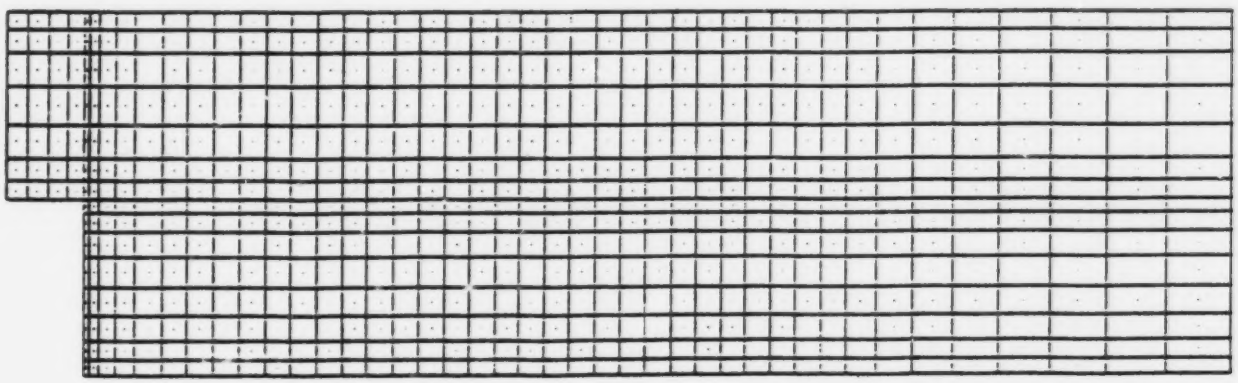
Figure 8. Configuration, coordinates, and nomenclature of backward-facing step flow; h: step height (0.0049 cm).

For the Reynolds number less than approximately 450, there exists only one recirculation zone at the down-stream region of the backward-facing step. As the Reynolds number is increased beyond approximately 450, another recirculation zone appears at the top wall of the channel, the size of which grows further as the Reynolds number is increased. Experimental data showed that a third recirculation zone appears at the bottom wall for the Reynolds number greater than approximately 1000. As the Reynolds number is increased beyond approximately 600, the three-dimensional effect becomes so strong that comparison between the two-dimensional computational result and the experimental data becomes less meaningful, which is discussed in detail in Armaly et al. [20]. Therefore, the computations have been carried out up to Re = 900 starting from Re = 100, with the incremental Reynolds number of 100.

The two different grids used are shown in Figure 9. For the coarse grid (Grid A) case, the uniform zero values for both velocity and pressure were used as an initial guess for the Re = 100 case, the converged solution of Re = 100 case was used as an initial guess for the Re = 200 case, and so on. The required number of iterations were 20, 24, 33, 43, 56, 65, 72, 80, and 77 for the Reynolds numbers of 100 through 900, respectively. For the fine grid case (Grid B), the initial guess for all the Reynolds number cases were obtained by interpolating the coarse grid solutions using the multi-grid concept of Ghia et al. [17].



(a)



(b)

Figure 9. Discretization of the backward-facing step flow. (a) Grid A, 51x31 grids (25x15 quadratic elements). (b) Grid B, 89x31 grids (44x15 quadratic elements).

The computed velocity vectors and the streamline contours for Re = 400, 500, and 800 are shown in Figures 10 and 11, respectively. The streamline contour labels are given in Table 4. The onset of the second recirculation zone at the top wall can be seen in the streamline contour plot given in Figure 11(a), yet there exists no recirculation zone for Re = 400 as can be confirmed from the velocity vector plot in Figure 10(a).

The computed static pressure was normalized using a relationship given as $P = pL_{ref}/V_{ref}/\mu$, where $L_{ref} = 0.0049$ m is the step height and V_{ref} is the bulk velocity at the inlet. The normalized pressure contours for Reynolds numbers of 400, 500, and 800 are shown in Figure 12.

The size of recirculation zones versus Reynolds number is compared with experimental data in Figure 13. It can be seen that the computational results compare favorably with experimental data up to Reynolds number of approximately 600.

The top and bottom wall pressure for the Reynolds number of 100 through 900 are shown in Figure 14. Neither experimental data nor computational results for the wall pressure are available as yet, and no comparison could have been made.

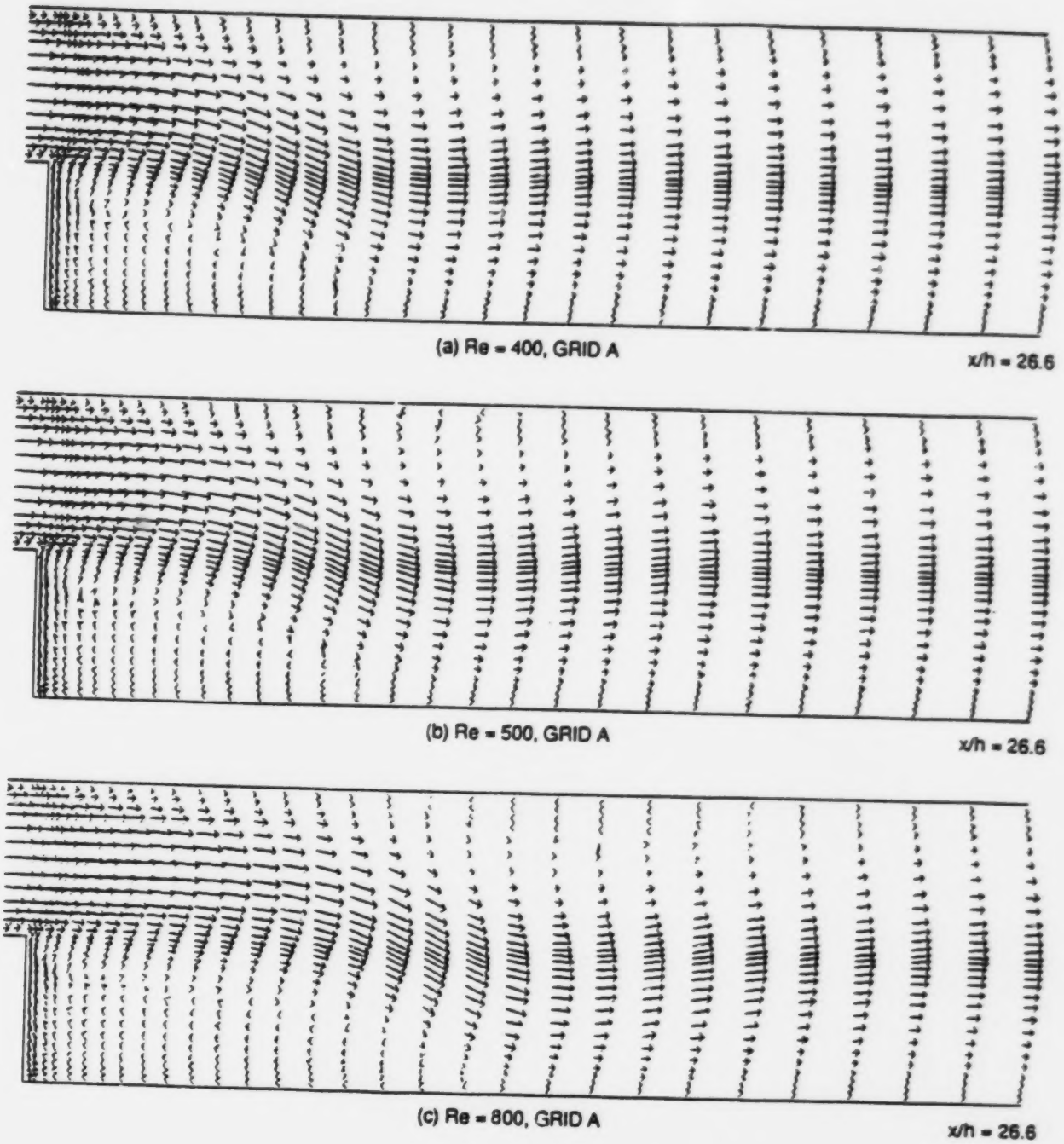


Figure 10. Velocity vectors for backward-facing step flow.

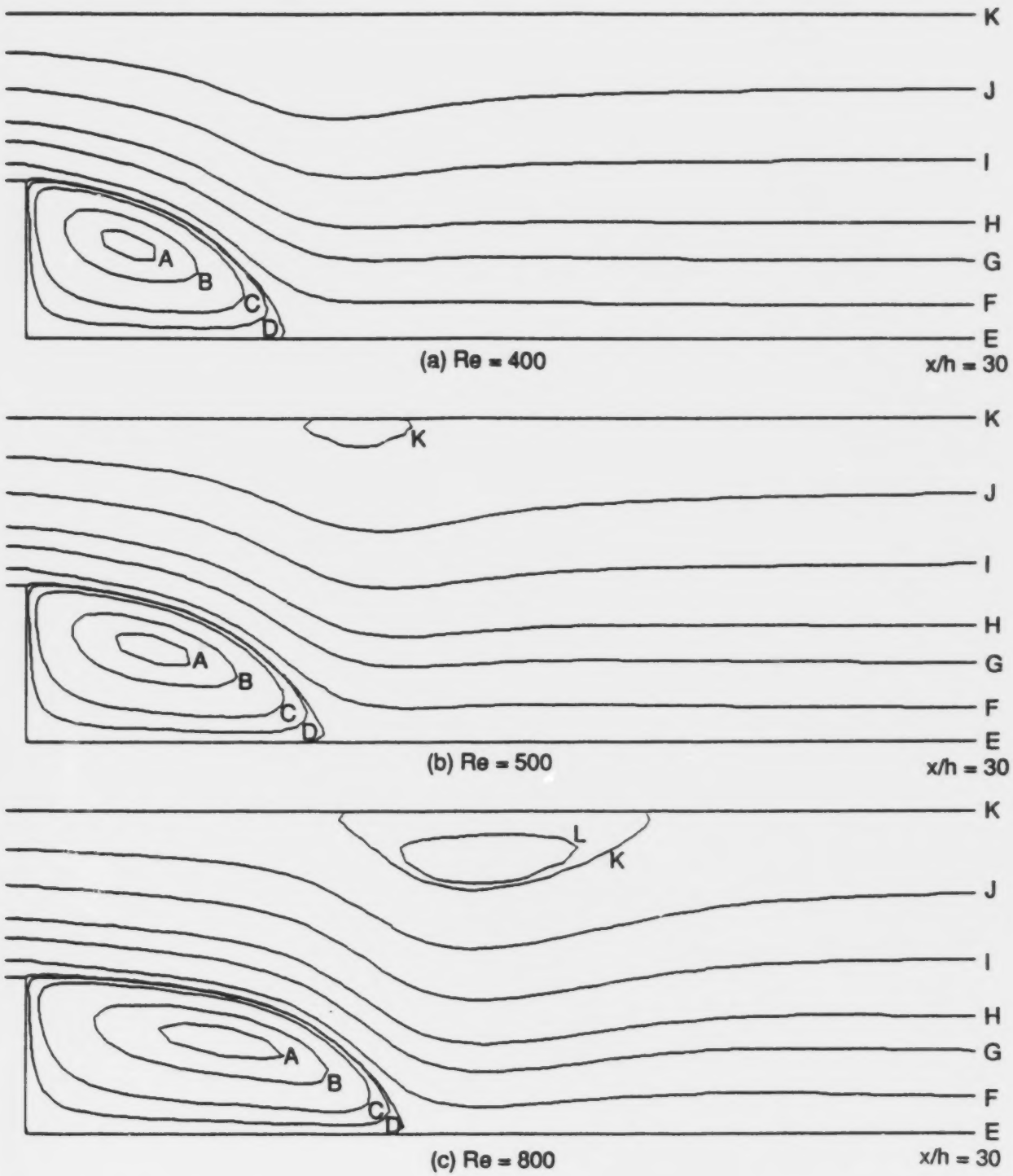


Figure 11. Streamlines for the backward-facing step flow.

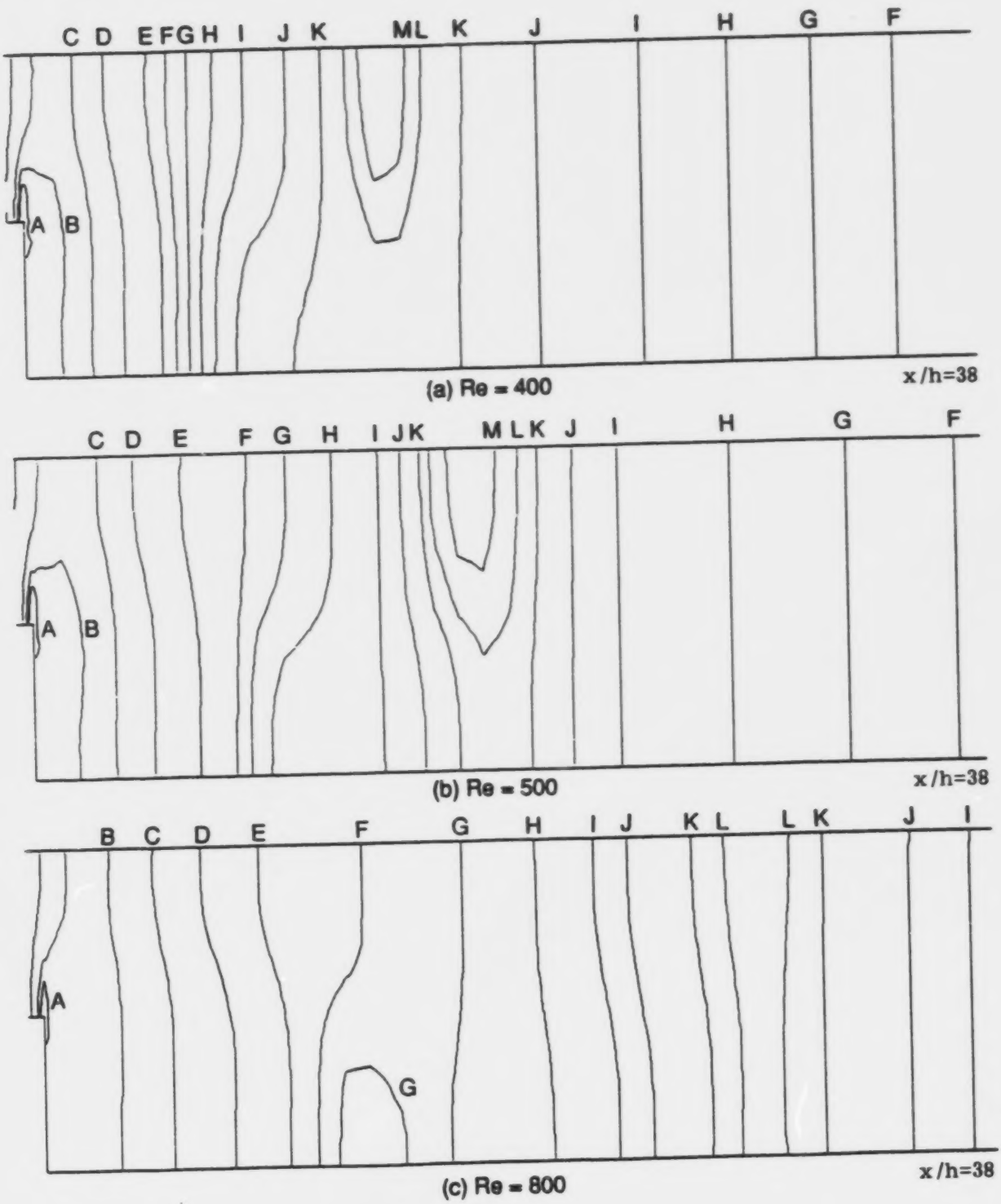


Figure 12. Pressure contours for the backward-facing step flow.

TABLE 4. STREAMLINE CONTOUR LABEL FOR BACKWARD-FACING STEP FLOW

Label	ψ	Label	ψ	Label	ψ
A	-2.0×10^{-4}	F	1.0×10^{-4}	K	3.467×10^{-3}
B	-1.5×10^{-4}	G	5.0×10^{-4}	L	3.480×10^{-3}
C	-5.0×10^{-5}	H	1.0×10^{-3}	M	3.50×10^{-3}
D	-1.0×10^{-5}	I	2.0×10^{-3}		
E	0.	J	3.0×10^{-3}		

TABLE 5. PRESSURE CONTOUR LABEL FOR BACKWARD-FACING STEP FLOW

Label	Reynolds Number (Re)		
	400.	500.	800.
A	-0.74	-0.72	-0.72
B	0.36	0.45	0.93
C	2.35	2.84	4.66
D	6.68	8.32	13.32
E	15.90	20.01	32.02
F	20.78	35.33	52.97
G	25.44	41.24	65.98
H	30.11	47.12	77.22
I	34.77	51.81	82.95
J	39.65	53.30	84.81
K	42.40	54.17	86.68
L	43.46	54.54	87.06
M	43.72	54.84	---

Finite difference computations of the same backward-facing step flow can be found in Kim and Moin [21] and Kwak and Chang [22] among many others. The same level of agreement as the present case between the computational results and the experimental data can be found in Kim and Moin [21]. On the other hand, the top wall recirculation zone was not shown as clearly as in the present result in the streamline contour plot due to Kwak and Chang [22].

Comparison of the present computational results with those of other investigators for a backward-facing step flow with the expansion ratio of 1:2 can be found in Reference 14.

3.3 Laminar Flow in a Square Duct of Strong Curvature

A re-developing laminar flow in a square duct of strong curvature is considered below. The flow configuration is shown in Figure 15, and the experimental data can be found in Humphery et al. [24].

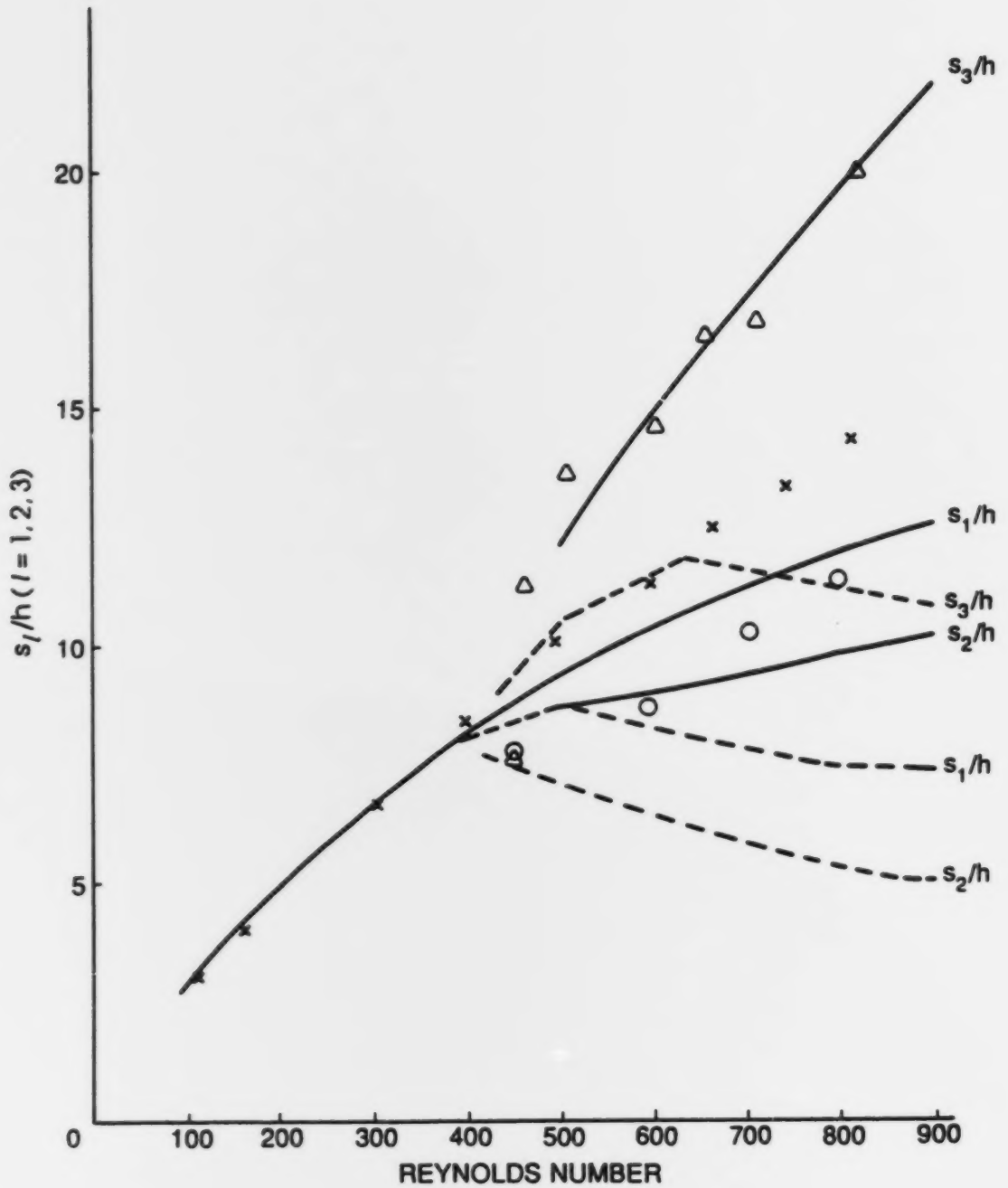
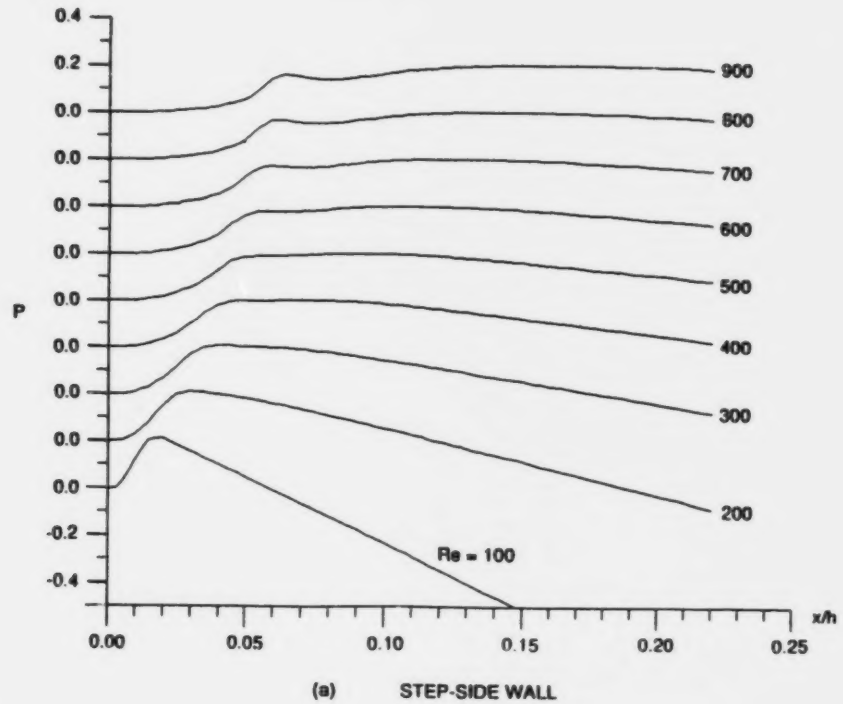
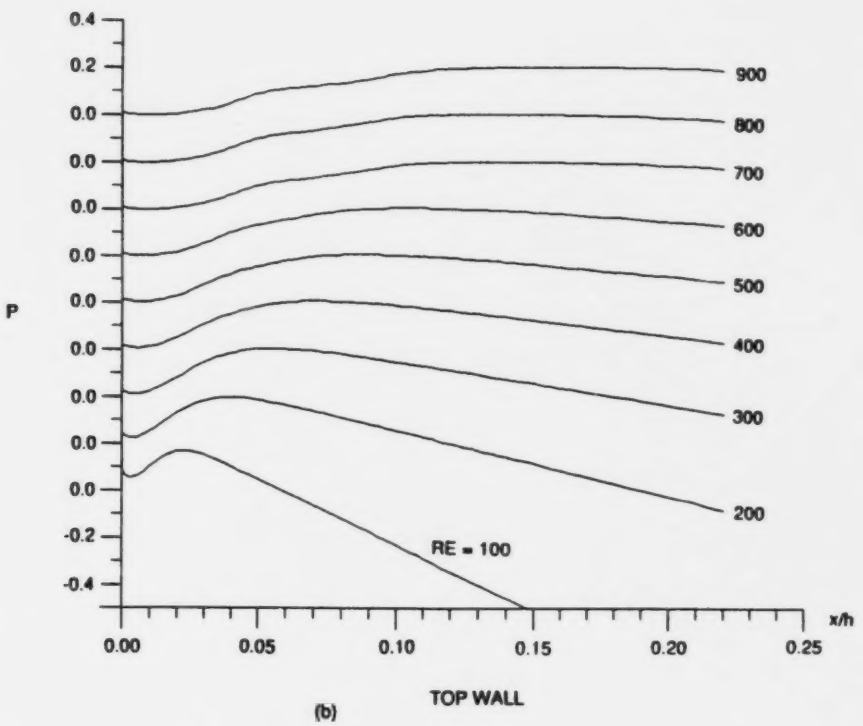


Figure 13. Reattachment length versus Reynolds number. ——: Present method; ---: control-volume based finite difference method [20]; x, O, Δ: Exp't s_1, s_2, and s_3, respectively.



(a) STEP-SIDE WALL



(b) TOP WALL

Figure 14. Wall pressure for backward-facing step flow, fine grid (Grid B) solutions, Re = 100 to 900.

The Reynolds number based on the hydraulic diameter of 0.04 m and the bulk velocity of 1.98×10^{-2} m/sec was 790. Due to the symmetry of the flow domain, only one half of the duct has been considered in the numerical analysis. The inlet plane has been located at 2.8 hydraulic diameters upstream of the curved section, and the exit plane, at 8 hydraulic diameters downstream of the end of the curved section. Discretization of the domain is shown in Figure 16. An analytical solution for the fully developed duct flow has been prescribed at the inlet boundary; and the vanishing normal stress has been prescribed at the exit boundary.

The computed velocity vectors at the three planes of the curved section are shown in Figure 17. It can be seen in Figure 17(c) that there exists a weak recirculation zone extending up to $\theta = 40°$ of the curved section. Computational results due to Humphery et al. [24] showed that the same recirculation zone extended beyond $\theta = 12°$, and the computational results due to Rhie [25] showed that the recirculation zone extended beyond $\theta = 30°$. There does not exist fine grid computational results for the recirculation zone as yet. The secondary recirculation flows at three cross-sections are shown in Figure 18. The computational results showed that the grid used was not fine enough to resolve all the details of the flow field. It was also found that the frontal solver used in this study was not adequate to pursue further grid refinement.

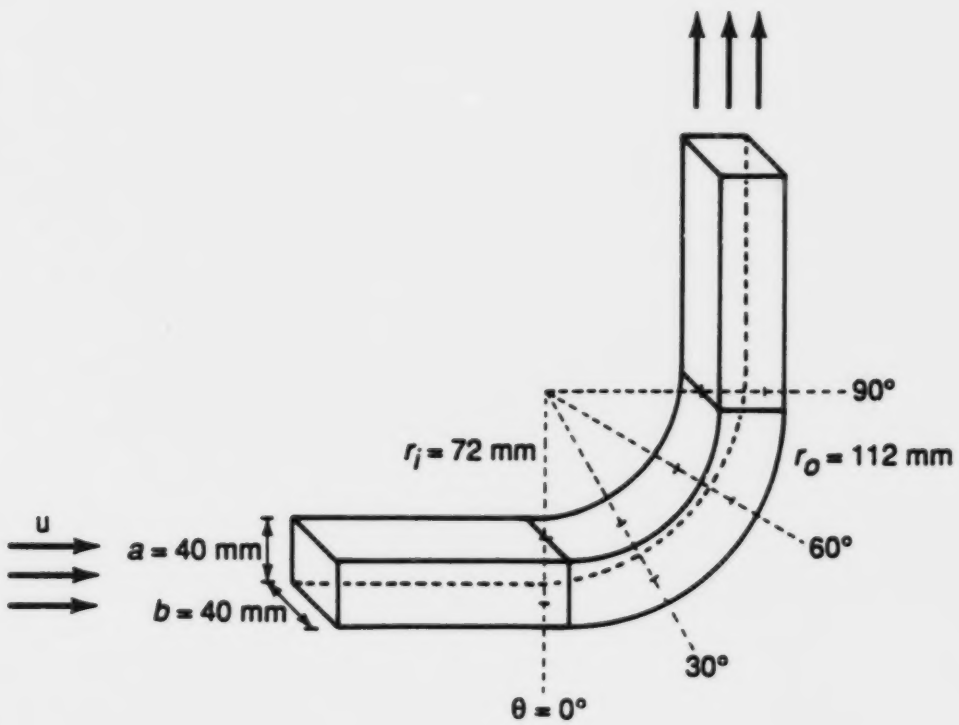


Figure 15. Configuration of the laminar flow in a square duct of strong curvature.

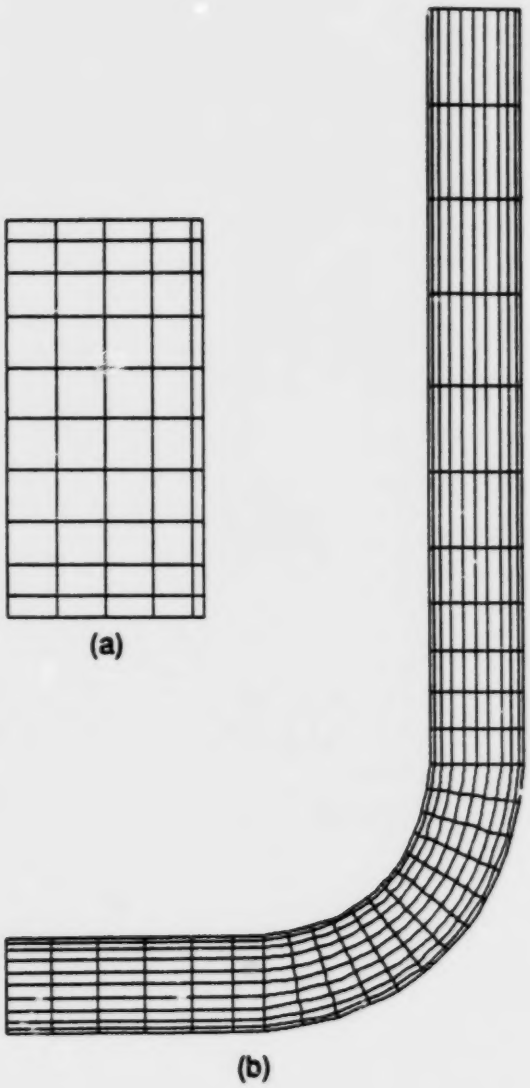


Figure 16. Discretization of the flow domain. (a) Discretization of the cross-section, and (b) grid in the flow direction.

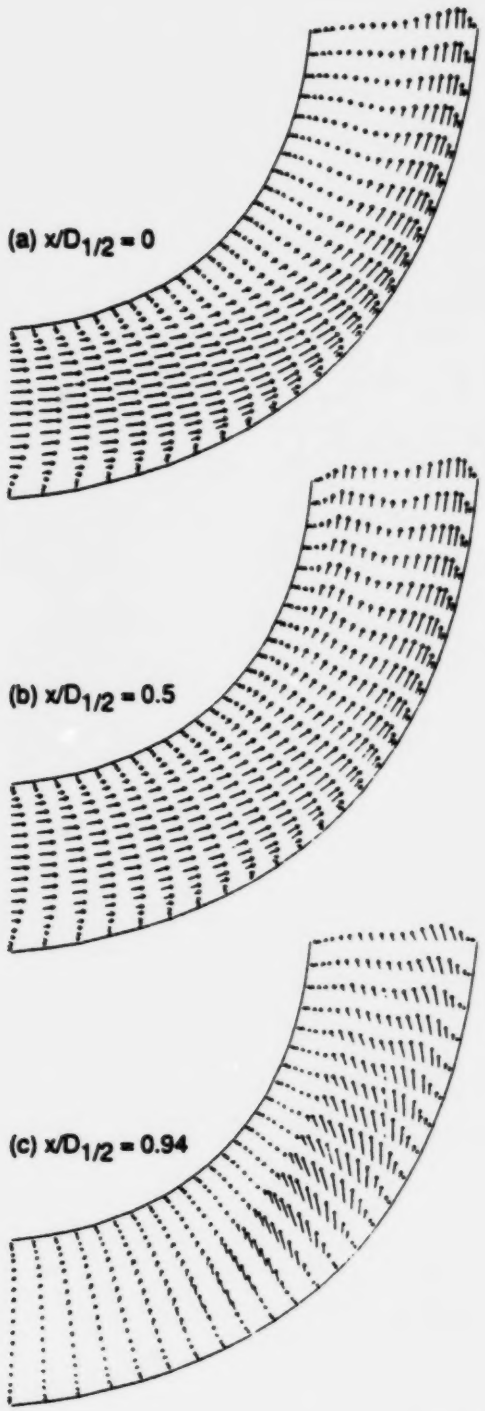
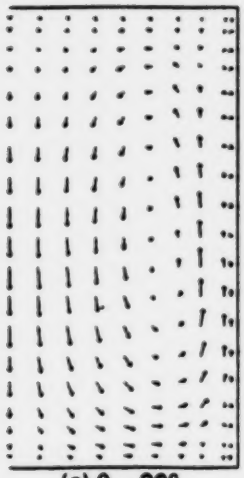
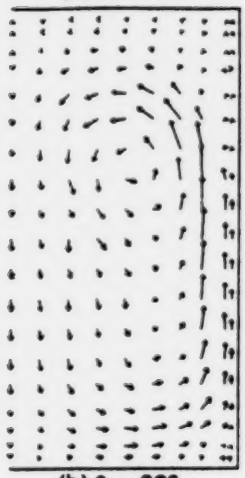


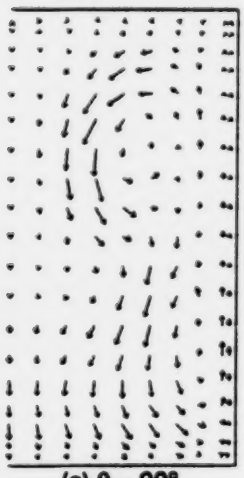
Figure 17. Velocity vectors on the curved section.



(a) θ = 30°



(b) θ = 60°



(c) θ = 90°

Figure 18. Secondary recirculation flows.

IV. CONCLUSIONS AND DISCUSSION

A velocity-pressure integrated, mixed interpolation, Galerkin finite element method for the Navier-Stokes equations has been presented. The method has been tested for high Reynolds number cavity flows, a backward-facing step flow, and a laminar flow in a square duct of strong curvature.

For the cavity flows, the present computational results compared favorably with those of Schreiber and Keller [16] and Ghia et al. [17], which were obtained by using approximately four times more grids than the present case.

For the backward-facing step flow, it has been shown that the present computational method could capture the subtle pressure driven recirculation zone at the top wall of the channel for Reynolds numbers greater than 500. The size of the recirculation zone also compared favorably with the experimental data.

For the three-dimensional curved duct flow, grid refinement was not feasible due to the computer limitation and the frontal solver used in the present study.

It is usually known that the Bubnov-Galerkin method [26] (i.e., the test functions are selected from the same space as that of the trial functions, as in the present case) yields oscillatory solutions for high Reynolds number flows. An extensive discussion on the finite element upwinding technique can be found in Brooks and Hughes [26], among many others. For the example problems considered herein, no upwinding was necessary to obtain accurate solutions which were free of numerical wiggles.

REFERENCES

1. Taylor, C. and Hughes, T. G.: Finite Element Programming of the Navier-Stokes Equation. Pineridge Press, Swansea, U.K., 1980.

2. Zienkiewicz, O. C., Taylor, R. L., and Baynham, J. M. W.: Mixed and Irreducible Formulations in Finite Element Analysis. Eds. S. N. Atluri, et al., Hybrid and Mixed Finite Element Methods, J. Wiley and Sons, New York, 1983.

3. Engelman, M. S., Sani, R. L., Gresho, P. M., and Bercovier, M.: Consistent vs. Reduced Integration Penalty Methods for Incompressible Media Using Several Old and New Elements. Int. J. Nume. Meth. Fluids, Vol. 2, 1982, pp. 25-42.

4. Kikuchi, N., Oden, J. T., and Song, Y. J.: Convergence of Modified Penalty Methods and Smoothing Schemes of Pressure for Stokes' Flow Problems. Eds. R. H. Gallagher, et al., Finite Elements in Fluids, Vol. 5, J. Wiley and Sons, New York, 1984, pp. 107-126.

5. Heinrich, J. C. and Marshall, R. S.: Viscous Incompressible Flow by Penalty Function Finite Element Method. Computers and Fluids, Vol. 9, 1981, pp. 73-83.

6. Patankar, S. V.: Numerical Heat Transfer and Fluid Flow. McGraw-Hill, New York, 1980.

7. Comini, G. and Giudice, S. D.: Finite Element Solution of the Incompressible Navier-Stokes Equations. Numerical Heat Transfer, Vol. 5, 1982, pp. 463-478.

8. Benim, A. C. and Zinser, W.: A Segregated Formulation of Navier-Stokes Equations with Finite Elements. Comput. Meth. Appl. Mech. Engrg., Vol. 57, 1986, pp. 223-237.

9. Rice, J. G. and Schnipke, R. J.: An Equal-Order Velocity-Pressure Formulation That Does Not Exhibit Spurious Pressure Modes. Comput. Meth. Appl. Mech. Engrg., Vol. 58, 1986, pp. 135-149.

10. Gallagher, R. G., et al., Eds.: Finite Element in Fluids. Vol. 1-6, J. Wiley and Sons, New York.

11. Oden, J. T. and Reddy, J. N.: An Introduction to the Mathematical Theory of Finite Elements. J. Wiley and Sons, New York, 1976.

12. Irons, B. and Ahmad, S.: Techniques of Finite Elements. J. Wiley and Sons, New York, 1980.

13. Dhatt, G. and Touzot, G.: The Finite Element Method Displayed. Translated by G. Cantin, J. Wiley and Sons, New York, 1984.

14. Kim. S.-W.: A Fine Grid Finite Element Computation of Two-Dimensional High Reynolds Number Flows. To appear in Computers and Fluids, 1988.

15. Burggraf, O. R.: Analytical and Numerical Studies of the Structure of Steady Separated Flows. J. Fluid Mech., Vol. 24, 1966, pp. 113-151.

16. Schreiber, R. and Keller, H. B.: Driven Cavity Flows by Efficient Numerical Techniques. J. Comput. Physics, Vol. 49, 1983, pp. 310-333.

17. Ghia, U., Ghia, K. N., and Shin, C. T.: High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. J. Comput. Physics, Vol. 48, 1982, pp. 387-411.

18. Gresho, P. M., Chan, S. T., Lee, R. L., and Upson, C. D.: A Modified Finite Element Method for Solving the Time-Dependent Incompressible Navier-Stokes Equations, Part 2. Applications. Int. J. Nume. Meth. Fluids, Vol. 4, 1984, pp. 619-640.

19. Bercovier, M. and Engelman, M.: A Finite Element for Numerical Solution of Viscous Incompressible Flows. J. Comput. Physics, Vol. 30, 1979, pp. 181-201.

20. Armaly, B. F., Durst, F., Pereira, J. C. F., and Schonung, B.: Experimental and Theoretical Investigation of Backward-Facing Step Flow. J. Fluid Mech., Vol. 127, 1983, pp. 473-496.

21. Kim, J. and Moin, P.: Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations. J. Comput. Physics, Vol. 59, 1985, pp. 308-323.

22. Kwak, D. and Chang, J. L. C.: A Three-Dimensional Incompressible Navier-Stokes Flow Solver, Part I. INS3D Code. CFC Workshop, University of Tennessee Space Institute, Tullahoma, Tennessee, June 1985.

23. Morgan, K., Periaux, J., and Thomasset, F., Eds.: Analysis of Laminar Flow Over a Backward-Facing Step. A GAMM-Workshop, Friedr Vieweg & Sohn, Germany, 1984.

24. Humphrey, J. A. C., Taylor, A. M. K., and Whitelaw, J. H.: Laminar Flow in a Square Duct of Strong Curvature. J. Fluid Mech., Vol. 83, 1977, pp. 509-527.

25. Rhie, C. M.: A Three-Dimensional Passage Flow Analysis Method Aimed at Centrifugal Impellers. Computers and Fluids, Vol. 13, No. 4, 1985, pp. 443-460.

26. Brooks, A. N., and Hughes, T. J. R.: Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equation. Comput. Meth. Applied Mech. Engrg., Vol. 32, 1982, pp. 199-259.

27. Zienkiewicz, O. C.: The Finite Element Method. McGraw-Hill, New York, 1977.

APPENDIX I

Finite Element Computer Program (NSFLOW/L) for Incompressible, Laminar Flows

PRECEDING PAGE BLANK NOT FILMED

PAGE 34 INTENTIONALLY BLANK

```
C
C********1*********2*********3*********4*********5*********6***
C     PROGRAM MAIN
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE,IWORD
      DIMENSION TITLE(15),IWORD(10)
      DATA IWORD / 'INIT',  'PREP',  '****',  'PROC',  'CONT',
     -             '****',  '****',  '****',  '****',  'END '/
      DATA MAXNOD,MAXELM,MAXDOF,MXFRON /4227, 1027, 11527, 167/
C
  101 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,601) TITLE
  501 FORMAT(20A4)
  601 FORMAT(/2X,20A4)
C
  102 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,601) TITLE
      DO 103 K=1,10
      IF(TITLE(2).EQ.IWORD(K)) GO TO 105
  103 CONTINUE
  104 WRITE(6,602)
  602 FORMAT(2X,'TERMINATED IN MAIN PROGRAM FOR INPUT DATA ERROR')
      STOP
C
  105 CONTINUE
      GO TO (1,2,3,4,5,  6,7,8,9,10), K
C
C     INITIALIZE DIMENSIONED VARIABLES
C
    1 CONTINUE
      CALL INITAL(MAXNOD,MAXELM,MAXDOF,MXFRON)
      GO TO 102
C
C     PREPARE INPUT DATA
C
    2 CONTINUE
      CALL PREP(MAXNOD,MAXELM,MAXDOF,MXFRON)
      GO TO 102
C
    3 CONTINUE
      GO TO 104
C
C     FLOW SOLVER - MAIN PROCESSOR
C
    4 CONTINUE
      CALL PROCES(MAXNOD,MAXELM,MAXDOF,MXFRON)
      GO TO 101
C
    5 CONTINUE
      GO TO 101
C
    6 CONTINUE
```


PRECEDING PAGE BLANK NOT FILMED

PAGE 36 INTENTIONALLY BLANK

```
    7 CONTINUE
    8 CONTINUE
    9 CONTINUE
   10 CONTINUE
      STOP
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE INITAL(MAXNOD,MAXELM,MAXDOF,MXFRON)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
C
      DO 10 KDIM=1,3
      DO 10 KNODE=1,MAXNOD
      X(KNODE,KDIM)    = 0.
   10 CONTINUE
C
      DO 30 KELEM=1,MAXELM
      DO 30 KPE=1,27
      NODES(KPE,KELEM)=0
   30 CONTINUE
C
      DO 40 KPROB=1,10
      DO 40 KNODE=1,MAXNOD
      IBCA(KNODE,KPROB) = 0
      ADBC(KNODE,KPROB) = 0.
      A(KNODE,KPROB)= 0.
   40 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      BLOCK DATA BLKDAT
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CLSCF/ XKSNOD(27,3,11),TM(4,4)
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CPROB/ IA(10),IPLOT
C
      DATA CLXKS(1,1)         /0./,
     -     CLW(1,1)           /2./,
     -     (CLXKS(K,2),K=1,2) /-0.5773502692, 0.5773502692/,
     -     (CLW(K,2),K=1,2)   / 1.,           1./,
     -     (CLXKS(K,3),K=1,3) /-0.7745966692, 0., 0.7745966692/,
     -     (CLW(K,3),K=1,3)/0.5555555556,0.8888888889,0.5555555556/,
     -     (CLXKS(K,4),K=1,4) /-0.8611363116, -0.3399810436,
```

```
     -                                0.3399810436,  0.8611363116/,
     -     (CLW(K,4),K=1,4)    / 0.3478548451,  0.6521451549,
     -                            0.6521451549,  0.3478548451/
C
      DATA (XKSNOD(K,1,6),K=1,4) / -1., 1., 1., -1./,
     -     (XKSNOD(K,2,6),K=1,4) / -1., -1., 1., 1./,
     -     (XKSNOD(K,1,8),K=1,9) / -1.,0.,1.,1.,1.,0.,-1.,-1.,0./,
     -     (XKSNOD(K,2,8),K=1,9) / -1.,-1.,-1.,0.,1.,1.,1.,0.,0./,
     -     (XKSNOD(K,1,11),K=1,27)/-1.,0.,1.,1.,1.,0.,-1.,-1.,
     -                             -1.,0.,1.,1.,1.,0.,-1.,-1.,
     -                             -1.,0.,1.,1.,1.,0.,-1.,-1.,
     -                              0.,0.,0./,
     -     (XKSNOD(K,2,11),K=1,27)/-1.,-1.,-1.,0.,1.,1.,1.,0.,
     -                             -1.,-1.,-1.,0.,1.,1.,1.,0.,
     -                             -1.,-1.,-1.,0.,1.,1.,1.,0.,
     -                              0.,0.,0./,
     -     (XKSNOD(K,3,11),K=1,27)/-1.,-1.,-1.,-1.,-1.,-1.,-1.,-1.,
     -                              0., 0., 0., 0., 0., 0., 0., 0.,
     -                              1., 1., 1., 1., 1., 1., 1., 1.,
     -                             -1., 1., 0./
C
C --- IFLOW=2 FOR 2-D INTEGRATED METHOD ---
      DATA (INDXF(KPE,1,2),KPE=1,9) / 1, 3, 5, 7, 9,11,13,15,17/,
     -     (INDXF(KPE,2,2),KPE=1,9) / 2, 4, 6, 8,10,12,14,16,18/,
     -     (INDXP(KPRE,2),KPRE=1,3) /19,20,21/
C --- IFLOW=3 FOR 2-D INTEGRATED METHOD ---
      DATA (INDXF(KPE,1,3),KPE=1,9) / 1, 3, 5, 7, 9,11,13,15,17/,
     -     (INDXF(KPE,2,3),KPE=1,9) / 2, 4, 6, 8,10,12,14,16,18/,
     -     (INDXP(KPRE,3),KPRE=1,3) /19,20,21/
C --- IFLOW=11 FOR 3-D INTEGRATED METHOD ---
      DATA (INDXF(KPE,1,11),KPE=1,27)
     -       /1, 4, 7,10,13,16,19,22,25,28,31,34,37,40,43,46,49,52,
     -        55,58,61,64,67,70,73,76,79/,
     -     (INDXF(KPE,2,11),KPE=1,27)
     -       /2, 5, 8,11,14,17,20,23,26,29,32,35,38,41,44,47,50,53,
     -        56,59,62,65,68,71,74,77,80/,
     -     (INDXF(KPE,3,11),KPE=1,27)
     -       /3, 6, 9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,
     -        57,60,63,66,69,72,75,78,81/,
     -     (INDXP(KPRE,11),KPRE=1,4)  /82,83,84,85/
C
      DATA ((TM(I,J),J=1,4),I=1,4) / 0.25, 0.25,  0.25, 0.25,
     -    -0.433012701,-0.433012701, 0.433012701, 0.433012701,
     -    -0.433012701, 0.433012701,-0.433012701, 0.433012701,
     -     0.75,       -0.75,       -0.75,        0.75/
C
      DATA IFLOW,IAXSY,IPLOT,NPRE,MGAUS,NGAUS,MAXIT /7*0/
      DATA VISCY,DENSY,PECLET /3*0./
      DATA (IA(K),K=1,10) /10*0/
      DATA ERROF/10*0./
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE DATLIB
```

```
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CPROB/ IA(10),IPLOT
      DIMENSION LIBELF(15),LIBNPE(11),LBNPRE(15)
C
      DATA (LIBELF(IFL),IFL=1,15) /0,8,8,0,0, 0,0,0,0,0, 11,0,0,0,0/,
     -     (LIBNPE(IEL),IEL=1,11) /0,0,0,0,0, 4,8,9,0,0, 27/,
     -     (LBNPRE(IFL),IFL=1,15) /0,3,3,0,0, 0,0,0,0,0, 4,0,0,0,0/
C
      IELF=LIBELF(IFLOW)
      NPRE=LBNPRE(IFLOW)
      NPE =LIBNPE(IELF)
C
      LGAUS=0
      MGAUS=NGAUS**NDIM
      GO TO (10,20,30), NDIM
   10 WRITE(6,601) NDIM
      STOP
  601 FORMAT(2X,'TERMINATED AT SUB-DATLIB    NDIM=',I2)
C
   20 CONTINUE
      DO 2 I=1,NGAUS
      DO 2 J=1,NGAUS
      LGAUS=LGAUS+1
      EXKS(1,LGAUS)=CLXKS(I,NGAUS)
      EXKS(2,LGAUS)=CLXKS(J,NGAUS)
      EW(LGAUS)    =CLW(I,NGAUS)*CLW(J,NGAUS)
    2 CONTINUE
      GO TO 100
C
   30 CONTINUE
      DO 3 I=1,NGAUS
      DO 3 J=1,NGAUS
      DO 3 K=1,NGAUS
      LGAUS=LGAUS+1
      EXKS(1,LGAUS)=CLXKS(I,NGAUS)
      EXKS(2,LGAUS)=CLXKS(J,NGAUS)
      EXKS(3,LGAUS)=CLXKS(K,NGAUS)
      EW(LGAUS)=CLW(I,NGAUS)*CLW(J,NGAUS)*CLW(K,NGAUS)
    3 CONTINUE
C
  100 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE PREP(MAXNOD,MAXELM,MAXDOF,MXFRON)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4    TITLE,ICNTRL,IWORD
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
```

```
     -                IAXSY,IELF
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CFRON/ MFRONF
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CPROB/ IA(10),IPLOT
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      COMMON /CWIND/ WHI(27),DWHX(27,3)
      DIMENSION TITLE(15),IWORD(30)
      DATA IWORD / 'DESC',  'CNTL',  'ELEM',  'NODE',  'MATE',
     -             '****',  '****',  'ITER',  '****',  '****',
     -             'IA01',  'IA02',  'IA03',  'IA04',  'IA05',
     -             'IA06',  'IA07',  'IA08',  'IA09',  'IA10',
     -             '****',  'EXAM',  '****',  '****',  '****',
     -             '****',  'INCL',  '****',  '****',  'END '/
C
  101 CONTINUE
      READ(5,501)  ICNTRL,TITLE
      WRITE(6,601) ICNTRL,TITLE
  501 FORMAT(20A4)
  601 FORMAT(/2X,20A4)
C
      DO 102 K=1,30
      IF(ICNTRL.EQ.IWORD(K)) GO TO 105
  102 CONTINUE
  103 WRITE(6,602)
      WRITE(6,601) ICNTRL,TITLE
  602 FORMAT(2X,'TERMINATED IN SUB-PREP FOR INPUT DATA ERROR')
      STOP
C
  105 CONTINUE
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
     -       21,22,23,24,25,26,27,28,29,30), K
C
    1 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,603) TITLE
      READ(5,*)    IFLOW,NDIM,NGAUS,MFRONF
      WRITE(6,605) IFLOW,NDIM,NGAUS,MFRONF
      IF(MFRONF.GT.MXFRON) GO TO 103
  603 FORMAT(2X,20A4)
  605 FORMAT(4X,'IFLOW=',I2,   2X,'NDIM =',I2,
     -       2X,'NGAUS=',I2,   2X,'MFRONF=',I5)
C
      CALL DATLIB
      WRITE(6,606) IELF,NPE,NPRE,MGAUS
  606 FORMAT(4X,'IELF=',I2,  2X,'NPE=',I2,  2X,'NPRE=',I2,
     -       2X,'MGAUS=',I2)
      WRITE(6,607)
      DO 40 LGAUS=1,MGAUS
```

```
      WRITE(6,608) (EXKS(KDIM,LGAUS),KDIM=1,NDIM),EW(LGAUS)
   40 CONTINUE
  607 FORMAT(4X,'NUMERICAL QUADRATURE DATA  EXKS AND EW')
  608 FORMAT(5X,4E12.4)
      GO TO 101
C
    2 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,603) TITLE
      READ(5,*) NNODE,NELEM,IAXSY,IPLOT
      WRITE(6,610) NNODE,NELEM,IAXSY,IPLOT
      IF(NNODE.GT.MAXNOD.OR.NELEM.GT.MAXELM) THEN
        WRITE(6,611) NNODE,NELEM,MAXNOD,MAXELM
        STOP
      ENDIF
  610 FORMAT(2X,'NNODE=',I5, 2X,'NELEM=',I5, 2X,'IAXSY=',I2,
     -       2X,'IPLOT=',I2)
  611 FORMAT(2X,'TERMINATED IN SUB-PREP FOR NNODE=',I6,
     -       2X,'NELEM=',I6,  2X,'MAXNOD=',I6,  2X,'MAXELM=',I6)
      GO TO 101
C
    3 CONTINUE
      CALL RELEM(NODES,NELEM,NPE,MAXELM)
      GO TO 101
C
    4 CONTINUE
      CALL RNODE(X,NNODE,NPE,IELF,NDIM,MAXNOD)
      GO TO 101
C
    5 CONTINUE
      READ(5,501) TITLE
      WRITE(6,603) TITLE
      READ(5,*) VISCY,DENSY,(BFX(K),K=1,NDIM)
      WRITE(6,613) VISCY,DENSY,(BFX(K),K=1,3)
  613 FORMAT(2X,'VISCY=',E12.4,     2X,'DENSY=',E12.4,
     -      /2X,'BFX=',3E12.4)
      GO TO 101
C
    6 CONTINUE
      GO TO 101
C
    7 CONTINUE
      GO TO 101
C
    8 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,603) TITLE
      READ(5,*) MAXIT,(RELAX(K),K=1.10),(CNVCF(K),K=1,10)
      WRITE(6,626) MAXIT,(RELAX(K),K=1,10),(CNVCF(K),K=1,10)
  626 FORMAT(2X,'MAXIT=',I5,      /4X,'RELAX=',5E12.4,
     -      /8X,5E12.4,
     -      /4X,'CNVCF=',5E12.4,  /8X,5E12.4)
      GO TO 101
C
```

```
     9 CONTINUE
       GO TO 101
C
    10 CONTINUE
       GO TO 101
C
    11 CONTINUE
       CALL RINIT(A(1,1),NNODE,MAXNOD)
       CALL RBC1(ADBC(1,1),IBCA(1,1),MAXNOD,NNODE)
       GO TO 101
C
    12 CONTINUE
       CALL RINIT(A(1,2),NNODE,MAXNOD)
       CALL RBC1(ADBC(1,2),IBCA(1,2),MAXNOD,NNODE)
       GO TO 101
C
    13 CONTINUE
       CALL RINIT(A(1,3),NNODE,MAXNOD)
       CALL RBC1(ADBC(1,3),IBCA(1,3),MAXNOD,NNODE)
       GO TO 101
C
C--------1---------2---------3---------4---------5---------6---
    14 CONTINUE
       READ(5,*) PBCDAT,(IPNOD(K),K=1,2)
       WRITE(6,635) PBCDAT,(IPNOD(K),K=1,2)
   635 FORMAT(2X,'PRESSURE B.C. DATA  PBCDAT=',E12.4,
      -       2X,'(IPNOD(K),K=1,2)=',2I6)
       GO TO 101
C
    15 CONTINUE
       GO TO 101
C
    16 CONTINUE
       GO TO 101
C
    17 CONTINUE
       GO TO 101
C
    18 CONTINUE
       GO TO 101
C
    19 CONTINUE
       GO TO 101
C
    20 CONTINUE
       GO TO 101
C
    21 CONTINUE
       GO TO 101
C
    22 CONTINUE
       GO TO 101
C
    23 CONTINUE
```

```
      GO TO 101
C
   24 CONTINUE
      GO TO 101
C
   25 CONTINUE
      GO TO 101
C
   26 CONTINUE
      GO TO 101
C
C     INCLUDE RE-START DATA
C
   27 CONTINUE
      CALL FEMDAT(A,ADBC,X,PBCDAT,NODES,IBCA,IPNOD,NPE,NNODE,
     -            NELEM,MAXNOD,MAXELM)
      GO TO 101
C--------1---------2---------3---------4---------5---------6---
   28 CONTINUE
      GO TO 103
C--------1---------2---------3---------4---------5---------6---
   29 CONTINUE
      RETURN
C--------1---------2---------3---------4---------5---------6---
   30 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RNODE(X,NNODE,NPE,IELF,NDIM,MAXNOD)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION X(MAXNOD,3),DELX(197,3),XNOD(27,3),NKS(3),CXKS(3),
     -          PHI(27),DPHI(27,3),WHI(27),DWHI(27,3),TITLE(15)
C
      READ(5,501) TITLE
      WRITE(6,601) TITLE
  501 FORMAT(20A4)
  601 FORMAT(2X,15A4)
C
      READ(5,*) NBLOC
      WRITE(6,605) NBLOC
  605 FORMAT(2X,'SUB-RNODE          NBLOC=',I2)
C
      DO 7000 IBLOC=1,NBLOC
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) METHOD
      WRITE(6,606) METHOD
  606 FORMAT(4X,'GRID GENERATION METHOD=',I3)
      GO TO (1000,2000) METHOD
C
 1000 CONTINUE
```

```
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) NODG1,INCRX,INCRY,INCRZ
      WRITE(6,640) NODG1,INCRX,INCRY,INCRZ
  640 FORMAT(4X,I5,  2X,I3,  2X,I3,  2X,I3)
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      DO 10 KDIM=1,3
      READ(5,*) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      WRITE(6,642) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      NKS(KDIM)=NDAT
      IF(NDAT.GT.197) THEN
        WRITE(6,645)
        STOP
      ENDIF
   10 CONTINUE
  642 FORMAT(2X,'NDAT=',I5,  20(/4X,5F10.7))
  645 FORMAT(2X,'INPUT DATA ERROR FOR NDAT IN SUB-RNODE')
      LLINE=NKS(1)
      MLINE=NKS(2)
      NLINE=NKS(3)
C
      DO 15 KLINE=1,NLINE
      DO 15 JLINE=1,MLINE
      DO 15 ILINE=1,LLINE
      KNODE=NODG1+(ILINE-1)*INCRX+(JLINE-1)*INCRY+(KLINE-1)*INCRZ
      X(KNODE,1)=DELX(ILINE,1)
      X(KNODE,2)=DELX(JLINE,2)
      X(KNODE,3)=DELX(KLINE,3)
   15 CONTINUE
      GO TO 7000
C
 2000 CONTINUE
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      DO 22 KPE=1,NPE
      WRITE(6,610) KPE,(XNOD(KPE,KDIM),KDIM=1,NDIM)
   22 CONTINUE
  610 FORMAT(4X,'KPE=',I2, 2X,3E12.4)
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) NODG1,INCRX,INCRY,INCRZ
      WRITE(6,612) NODG1,INCRX,INCRY,INCRZ
  612 FORMAT(4X,I5,  2X,I3,  2X,I3,  2X,I3)
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      DO 25 KDIM=1,3
      READ(5,*) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      WRITE(6,614) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      NKS(KDIM)=NDAT
      IF(NDAT.GT.197) THEN
        WRITE(6,620)
        STOP
```

```
      ENDIF
   25 CONTINUE
  614 FORMAT(2X,'NDAT-',I5,  10(/4X,10F5.2))
  620 FORMAT(2X,'INPUT DATA ERROR FOR NDAT IN SUB-RNODE')
      LLINE-NKS(1)
      MLINE-NKS(2)
      NLINE-NKS(3)
C
      DO 45 KLINE-1,NLINE
      DO 45 JLINE-1,MLINE
      DO 45 ILINE-1,LLINE
      CXKS(1)-DELX(ILINE,1)
      CXKS(2)-DELX(JLINE,2)
      CXKS(3)-DELX(KLINE,3)
C
      GO TO (31,31,31,31,31,  36,37,38,31,31,  41), IELF
   31 CONTINUE
      WRITE(6,630) IELF
  630 FORMAT(2X,'SUB-RNODE    IELF -',I2)
      STOP
   36 CALL SHAP21(PHI,DPHI,CXKS,NPE)
      GO TO 42
   37 CALL SHAP22(PHI,DPHI,CXKS,NPE)
      GO TO 42
   38 CALL SHAP23(PHI,DPHI,CXKS,NPE)
      GO TO 42
   41 CALL SHAP33(PHI,DPHI,CXKS,NPE)
C
   42 CONTINUE
      KNODE-NODG1+(ILINE-1)*INCRX+(JLINE-1)*INCRY+(KLINE-1)*INCRZ
      DO 44 KDIM-1,NDIM
      X(KNODE,KDIM)-0.
      DO 44 KPE-1,NPE
      X(KNODE,KDIM)-X(KNODE,KDIM)+XNOD(KPE,KDIM)*PHI(KPE)
   44 CONTINUE
   45 CONTINUE
C
 7000 CONTINUE
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RELEM(NODES,NELEM,NPE,MAXELM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION NODES(27,MAXELM),NEL(3),INCREL(3),INCNOD(27,3),
     -          TITLE(15)
C
      DO 1 KELEM-1,NELEM
      DO 1 KPE-1,NPE
      NODES(KPE,KELEM)-0
    1 CONTINUE
C
      READ(5,501) TITLE
```

```
      WRITE(6,601) TITLE
  501 FORMAT(20A4)
  601 FORMAT(2X,15A4)
      READ(5,*) NBLOC
      WRITE(6,610) NBLOC
  610 FORMAT(4X,'NBLOC-',I6)
C
      DO 100 IBLOC-1,NBLOC
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*)    IEL1,(NODES(IPE,IEL1),IPE-1,NPE)
      WRITE(6,620) IEL1,(NODES(IPE,IEL1),IPE-1,NPE)
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      DO 10 KDIM-1,3
      READ(5,*) NEL(KDIM),INCREL(KDIM),(INCNOD(K,KDIM),K-1,NPE)
   10 CONTINUE
      NELX-NEL(1)
      NELY-NEL(2)
      NELZ-NEL(3)
C
      DO 50 IELZ-1,NELZ
      DO 50 IELY-1,NELY
      DO 50 IELX-1,NELX
      KELEM-IEL1+(IELX-1)*INCREL(1)+(IELY-1)*INCREL(2)
     -                              +(IELZ-1)*INCREL(3)
      DO 30 KPE-1,NPE
      NODES(KPE,KELEM)-NODES(KPE,IEL1)+(IELX-1)*INCNOD(KPE,1)
     -         +(IELY-1)*INCNOD(KPE,2)+(IELZ-1)*INCNOD(KPE,3)
   30 CONTINUE
   50 CONTINUE
C
  100 CONTINUE
  620 FORMAT(4X,'IEL1-',I6, 2X,'NODES(KPE,IEL1)-',8I6,
     -       3(/15X,'NODES(KPE,IEL1)-',8I6))
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RINIT(AINIT,NNODE,MAXNOD)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AINIT(MAXNOD),TITLE(15)
C
      READ(5,501) TITLE
      WRITE(6,601) TITLE
  501 FORMAT(15A4)
  601 FORMAT(/2X,15A4)
      READ(5,*) NREC
      WRITE(6,610) NREC
      IF(NREC.LE.0) RETURN
  610 FORMAT(2X,'SUB-RINIT     NREC-',I5)
C
      DO 20 IREC-1,NREC
```

```
      READ(5,*) N1,N2,INCNOD,ADATA
      WRITE(6,620) N1,N2,INCNOD,ADATA
  620 FORMAT(5X,'N1-',I6,  5X,'N2-',I6,  5X,'INCNOD-',I6,
     -       5X,'ADATA-',E12.4)
      DO 10 N-N1,N2,INCNOD
      AINIT(N)-ADATA
   10 CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RBC1(DBCH,LDBC,MAXNOD,NNODE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION DBCH(MAXNOD),LDBC(MAXNOD),TITLE(15)
C
      DO 10 KNODE-1,NNODE
      LDBC(KNODE)-0
      DBCH(KNODE)-0.
   10 CONTINUE
C
C     DBC DATA
C
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) NREC
      WRITE(6,602) NREC
      IF(NREC.EQ.0) RETURN
C
      WRITE(6,603)
      DO 40 IREC-1,NREC
      READ(5,*) N1,N2,INCR,DUM
      WRITE(6,604) N1,N2,INCR,DUM
      DO 30 K-N1,N2,INCR
      LDBC(K)-1
      DBCH(K)-DUM
   30 CONTINUE
   40 CONTINUE
C
      WRITE(6,607)
      DO 60 KNODE-1,NNODE
      IF(LDBC(KNODE).NE.0) WRITE(6,605) KNODE,LDBC(KNODE),
     -                                  DBCH(KNODE)
   60 CONTINUE
C
      RETURN
  501 FORMAT(20A4)
  601 FORMAT(/2X,20A4)
  602 FORMAT(5X,'NO. OF INPUT DATA RECORD FOR DBC, NREC-',I5)
  603 FORMAT(5X,'  N1-NODE    N2-NODE   INCREMENT    DBC-DATA')
  604 FORMAT(5X,I5,5X,I5,5X,I5,5X,E11.4)
  605 FORMAT(5X,'NODE-',I5,5X,'LDBC-',I3,5X,'DATDBC-',E11.4)
```

```
  607 FORMAT(/2X,'LIST OF D.B.C. DATA FROM SUB-RBC1')
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE FEMDAT(A,ADBC,X,PBCDAT,NODES,IBCA,IPNOD,NPE,
     -                  NNODE,NELEM,MAXNOD,MAXELM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION A(MAXNOD,10),ADBC(MAXNOD,10),X(MAXNOD,3),
     -          NODES(27,MAXELM),IBCA(MAXNOD,10),IPNOD(2),TITLE(15)
C
      READ(4,501) TITLE
  501 FORMAT(15A4)
  601 FORMAT(2X,15A4)
      DO 10 KNOD=1,NNODE
      READ(4,*) KNODE,(X(KNODE,KDUM),KDUM=1,3)
   10 CONTINUE
C
      READ(4,501) TITLE
      DO 20 KEL=1,NELEM
      READ(4,*) KELEM,(NODES(KPE,KELEM),KPE=1,NPE)
   20 CONTINUE
C
      READ(4,501) TITLE
      DO 30 KNOD=1,NNODE
      READ(4,*) KNODE,(IBCA(KNODE,K),K=1,3)
   30 CONTINUE
C
      READ(4,501) TITLE
      READ(4,*) (IPNOD(K),K=1,2),PBCDAT
C
      READ(4,501) TITLE
      DO 40 KNOD=1,NNODE
      READ(4,*) KNODE,(A(KNODE,K),K=1,4)
   40 CONTINUE
C
      READ(4,501) TITLE
      DO 57 KNOD=1,NNODE
      READ(4,*) KNODE,(ADBC(KNODE,K),K=1,3)
   57 CONTINUE
C
      DO 60 KNODE=1,NNODE
      A(KNODE,4)=0.
   60 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE ISOPEL(IFLOW,IELF,IAXSY,NPE,NPRE,NDIM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CELEM/ EX(27,3),EA(27,10),NODEL(27),IELEM
      COMMON /CWIND/ WHI(27),DWHX(27,3)
      COMMON /ESHAP/ PHI(27),DPHX(27,3),PSI(27),DXDK(3,3),
```

```
     -        DKDX(3,3),CXKS(3),DETJB,RADUS,IGAUS,JGAUS,LGAUS
      DIMENSION DPHI(27,3),DPSI(27,3),XGS(3)
C
      GO TO (1,1,1,1,1,  6,7,8,1,1,  11), IELF
    1 CONTINUE
      WRITE(6,608) IFLOW,IELF
  608 FORMAT(2X,'SUB-ISOPEL  IFLOW=',I2,  2X,'IELF=',I2)
      STOP
C
    6 CALL SHAP21(PHI,DPHI,CXKS,NPE)
      GO TO 15
C
    7 CALL SHAP22(PHI,DPHI,CXKS,NPE)
      GO TO 15
C
    8 CALL SHAP23(PHI,DPHI,CXKS,NPE)
      GO TO 15
C
   11 CALL SHAP33(PHI,DPHI,CXKS,NPE)
C
   15 CONTINUE
      DO 25 IDIM=1,NDIM
      DO 25 ICE=1,NDIM
      DXDK(IDIM,ICE)=0.
      DO 24 KPE=1,NPE
      DXDK(IDIM,ICE) = DXDK(IDIM,ICE)
     -               + EX(KPE,IDIM)*DPHI(KPE,ICE)
   24 CONTINUE
   25 CONTINUE
C
      GO TO (31,32,33), NDIM
   31 CONTINUE
      DETJB=DXDK(1,1)
      GO TO 37
   32 CONTINUE
      DETJB=DXDK(1,1)*DXDK(2,2)-DXDK(1,2)*DXDK(2,1)
      GO TO 37
   33 CONTINUE
      DETJB =  DXDK(1,1)*DXDK(2,2)*DXDK(3,3)
     -       + DXDK(1,2)*DXDK(2,3)*DXDK(3,1)
     -       + DXDK(2,1)*DXDK(3,2)*DXDK(1,3)
     -       - DXDK(1,1)*DXDK(2,3)*DXDK(3,2)
     -       - DXDK(2,2)*DXDK(1,3)*DXDK(3,1)
     -       - DXDK(3,3)*DXDK(2,1)*DXDK(1,2)
C
   37 CONTINUE
      IF(DETJB.LE.1.E-15) THEN
         WRITE(6,610) IELEM,IELF,NPE,NPRE
         DO 17 KPE=1,NPE
   17    WRITE(6,615) KPE,NODEL(KPE),(EX(KPE,IDIM),IDIM=1,NDIM)
         WRITE(6,620) (PHI(K),K=1,NPE)
         DO 18 IDIM=1,NDIM
         WRITE(6,625) (DPHI(K,IDIM),K=1,NPE)
   18    CONTINUE
```

```
      WRITE(6,630) DETJB,((DXDK(I,J),J=1,NDIM),I=1,NDIM)
      STOP
     ENDIF
 610 FORMAT(2X,'PROGRAM RUN TERMINATED AT SUB-ISOPEL DUE TO ',
     -          'SMALL DETJB',  /4X,'IELEM=',I5, 2X,'IELF=',I2,
     -       2X,'NPE=',I2,  2X,'NPRE=',I2)
 615 FORMAT(3X,'IPE =',I2, 2X,'INODE=',I5, 2X,'XDAT=',3E12.4)
 620 FORMAT(4X,'PHI =',5F10.4,  5(/5X,'PHI =',5F10.4))
 625 FORMAT(4X,'DPHI=',5F10.4,  5(/4X,'DPHI=',5F10.4))
 630 FORMAT(2X,'DETJB=',E11.4,/2X,'DXDK=',3(/5X,3E12.4))
C
     GO TO (41,42,43), NDIM
  41 CONTINUE
     DKDX(1,1)=1./DETJB
     GO TO 45
  42 CONTINUE
     DKDX(1,1)=DXDK(2,2)/DETJB
     DKDX(1,2)=-DXDK(1,2)/DETJB
     DKDX(2,1)=-DXDK(2,1)/DETJB
     DKDX(2,2)= DXDK(1,1)/DETJB
     GO TO 45
C
  43 CONTINUE
     DKDX(1,1)=(DXDK(2,2)*DXDK(3,3)-DXDK(3,2)*DXDK(2,3))/DETJB
     DKDX(1,2)=(DXDK(1,3)*DXDK(3,2)-DXDK(1,2)*DXDK(3,3))/DETJB
     DKDX(1,3)=(DXDK(1,2)*DXDK(2,3)-DXDK(2,2)*DXDK(1,3))/DETJB
     DKDX(2,1)=(DXDK(2,3)*DXDK(3,1)-DXDK(2,1)*DXDK(3,3))/DETJB
     DKDX(2,2)=(DXDK(1,1)*DXDK(3,3)-DXDK(3,1)*DXDK(1,3))/DETJB
     DKDX(2,3)=(DXDK(2,1)*DXDK(1,3)-DXDK(1,1)*DXDK(2,3))/DETJB
     DKDX(3,1)=(DXDK(2,1)*DXDK(3,2)-DXDK(2,2)*DXDK(3,1))/DETJB
     DKDX(3,2)=(DXDK(1,2)*DXDK(3,1)-DXDK(1,1)*DXDK(3,2))/DETJB
     DKDX(3,3)=(DXDK(1,1)*DXDK(2,2)-DXDK(2,1)*DXDK(1,2))/DETJB
C
  45 CONTINUE
C
C    CALCULATE GLOBAL DERIVATIVES
C
     DO 47 IDIM=1,NDIM
     DO 47 IPE=1,NPE
     DPHX(IPE,IDIM)=0.0
     DO 47 ICE=1,NDIM
     DPHX(IPE,IDIM) = DPHX(IPE,IDIM)
    -                 + DPHI(IPE,ICE)*DKDX(ICE,IDIM)
  47 CONTINUE
C
     IF(IFLOW.EQ.0) GO TO 85
     GO TO (51,52,53,85,85,  85,85,85,85,85,  61,85,85,85,85),
    -       IFLOW
C
  51 CALL SHAP01(PSI,DPSI,CXKS,NPRE)
     GO TO 85
C
  52 CALL SHAP02(PSI,DPSI,CXKS,NPRE)
     GO TO 85
```

```
C
   53 CONTINUE
      DO 57 KDIM=1,NDIM
      XGS(KDIM)=0.
      DO 57 KPE=1,NPE
      XGS(KDIM)=XGS(KDIM)+EX(KPE,KDIM)*PHI(KPE)
   57 CONTINUE
      PSI(1)=1.
      DO 58 KDIM=1,NDIM
   58 PSI(KDIM+1)=XGS(KDIM)
      GO TO 85
C
   61 CALL SHAP03(PSI,DPSI,CXKS,NPRE)
C
   [illegible]
      IF(IAXSY.EQ.1) THEN
        RADUS=0
        DO 90 KPE=1,NPE
        RADUS=RADUS+EX(KPE,2)*PHI(KPE)
   90   CONTINUE
      ENDIF
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE LSHP1(PLK,DPLK,S)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PLK(3),DPLK(3)
C
      PLK(1)=(1.-S)/2.
      PLK(2)=(1.+S)/2.
      DPLK(1)=-0.5
      DPLK(2)= 0.5
      RETURN
      END
C
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE LSHP2(PNK,DPNK,S)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PNK(3),DPNK(3)
C
C     1-D QUADRATIC ELEMENT  (NE= 1   2    3)
C                            (S =-1.  0.   1.)
      PNK(1)=S*(S-1.)/2.
      PNK(2)=1.-S**2
      PNK(3)=S*(1.+S)/2.
C
      DPNK(1)=S-0.5
      DPNK(2)=-2.*S
      DPNK(3)=S+0.5
      RETURN
      END
```

```
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP01(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),CXKS(3)
C
      SHP(1)=1.
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP02(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),CXKS(3)
      SHP(1)=0.333333333+0.816496582*CXKS(2)
      SHP(2)=0.333333333-0.707106781*CXKS(1)-0.408248291*CXKS(2)
      SHP(3)=0.333333333+0.707106781*CXKS(1)-0.408248291*CXKS(2)
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP03(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),CXKS(3)
      SHP(1)=0.25+0.612372435*CXKS(2)-0.433012701*CXKS(3)
      SHP(2)=0.25-0.612372435*CXKS(2)-0.433012701*CXKS(3)
      SHP(3)=0.25+0.612372435*CXKS(1)+0.433012701*CXKS(3)
      SHP(4)=0.25-0.612372435*CXKS(1)+0.433012701*CXKS(3)
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP21(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),CXKS(3)
C
C     4-NODE LINEAR ELEMENT COUNTER-CLOCKWISE NODE NUMBERING
C
      S=CXKS(1)
      T=CXKS(2)
      ST=S*T
      SHP(1)=0.25*(1.-S-T+ST)
      SHP(2)=0.25*(1.+S-T-ST)
      SHP(3)=0.25*(1.+S+T+ST)
      SHP(4)=0.25*(1.-S+T-ST)
C
      DSHP(1,1)=0.25*(-1.+T)
      DSHP(2,1)=0.25*( 1.-T)
      DSHP(3,1)=0.25*( 1.+T)
      DSHP(4,1)=0.25*(-1.-T)
      DSHP(1,2)=0.25*(-1.+S)
      DSHP(2,2)=0.25*(-1.-S)
      DSHP(3,2)=0.25*( 1.+S)
      DSHP(4,2)=0.25*( 1.-S)
```

```
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP22(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),CXKS(3)
C
      S=CXKS(1)
      T=CXKS(2)
      ST=S*T
      SS=S*S
      TT=T*T
      SST=SS*T
      STT=S*TT
      S2=2.*S
      T2=2.*T
      ST2=2.*ST
C
      SHP(1)=0.25*(-1.+ST+SS+TT-SST-STT)
      SHP(2)=0.50*( 1.- T-SS+SST)
      SHP(3)=0.25*(-1.-ST+SS+TT-SST+STT)
      SHP(4)=0.50*( 1.+S-TT-STT)
      SHP(5)=0.25*(-1.+ST+SS+TT+SST+STT)
      SHP(6)=0.50*( 1.+T-SS-SST)
      SHP(7)=0.25*(-1.-ST+SS+TT+SST-STT)
      SHP(8)=0.50*( 1.-S-TT+STT)
C
      DSHP(1,1)=0.25*( T+S2-ST2-TT)
      DSHP(2,1)=      -S+ST
      DSHP(3,1)=0.25*(-T+S2-ST2+TT)
      DSHP(4,1)=0.50*( 1.-TT)
      DSHP(5,1)=0.25*( T+S2+ST2+TT)
      DSHP(6,1)=      -S-ST
      DSHP(7,1)=0.25*(-T+S2+ST2-TT)
      DSHP(8,1)=0.50*(-1.+TT)
C
      DSHP(1,2)=0.25*( S+T2-SS-ST2)
      DSHP(2,2)=0.50*(-1.+SS)
      DSHP(3,2)=0.25*(-S+T2-SS+ST2)
      DSHP(4,2)=      -T-ST
      DSHP(5,2)=0.25*( S+T2+SS+ST2)
      DSHP(6,2)=0.50*( 1.-SS)
      DSHP(7,2)=0.25*(-S+T2+SS-ST2)
      DSHP(8,2)=      -T+ST
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP23(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),PNK(3),
     -          DPNK(3),PNE(3),DPNE(3),INDK(9),INDE(9),CXKS(3)
      DATA (INDK(KPE),KPE=1,9) /1,2,3,  3,3,2,  1,1,2/,
```

```
     -     (INDE(KPE),KPE=1,9) /1,1,1,  2,3,3,  3,2,2/
C                                7  6  5
C     9 NODE QUADRATIC ELEMENT   8  9  4
C                                1  2  3
      CALL LSHP2(PNK,DPNK,CXKS(1))
      CALL LSHP2(PNE,DPNE,CXKS(2))
C
      DO 10 KPE=1,NPE
      IPE=INDK(KPE)
      JPE=INDE(KPE)
      SHP(KPE)=PNK(IPE)*PNE(JPE)
      DSHP(KPE,1)=DPNK(IPE)*PNE(JPE)
      DSHP(KPE,2)=PNK(IPE)*DPNE(JPE)
   10 CONTINUE
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP33(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),PNK(3),
     -          DPNK(3),PNE(3),DPNE(3),PNZ(3),DPNZ(3),INDK(27),
     -          INDE(27),INDZ(27),CXKS(3)
      DATA (INDK(K),K=1,27),(INDE(K),K=1,27),(INDZ(K),K=1,27)
     -     /1,2,3,3,3,2,1,1, 1,2,3,3,3,2,1,1, 1,2,3,3,3,2,1,1, 2,2,2,
     -      1,1,1,2,3,3,3,2, 1,1,1,2,3,3,3,2, 1,1,1,2,3,3,3,2, 2,2,2,
     -      1,1,1,1,1,1,1,1, 2,2,2,2,2,2,2,2, 3,3,3,3,3,3,3,3, 1,3,2/
C
      CALL LSHP2(PNK,DPNK,CXKS(1))
      CALL LSHP2(PNE,DPNE,CXKS(2))
      CALL LSHP2(PNZ,DPNZ,CXKS(3))
C
      DO 10 K=1,NPE
      IPE=INDK(K)
      JPE=INDE(K)
      KPE=INDZ(K)
      SHP(K)=PNK(IPE)*PNE(JPE)*PNZ(KPE)
      DSHP(K,1)=DPNK(IPE)*PNE(JPE)*PNZ(KPE)
      DSHP(K,2)=PNK(IPE)*DPNE(JPE)*PNZ(KPE)
      DSHP(K,3)=PNK(IPE)*PNE(JPE)*DPNZ(KPE)
   10 CONTINUE
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE PROCES(MAXNOD,MAXELM,MAXDOF,MXFRON)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CFRON/ MFRONF
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CPROB/ IA(10),IPLOT
C
```

```
      CALL PFRONT(NODES,NNODE,NELEM,NPE,MAXELM)
      CALL SHPLIB
C
      CALL SFLOW(MAXNOD,MAXELM,MAXDOF,MXFRON)
      CALL PFLOW(MAXNOD,MAXELM,MAXDOF)
      IF(IPLOT.GE.1) CALL PLSDAT(MAXNOD,MAXELM,MAXDOF)
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE PFRONT(NODES,NNODE,NELEM,NPE,MAXELM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION NODES(27,MAXELM)
C
C     FIND LAST APPEARENCE OF EACH NODE AT FIRST ITERATION ONLY
C
      DO 30 INODE=1,NNODE
      LASTE=0
      DO 20 KELEM=1,NELEM
      DO 10 IPE=1,NPE
      INODP=ABS(NODES(IPE,KELEM))
      IF(INODP.NE.INODE) GO TO 10
      LASTE=KELEM
      LASTN=IPE
      GO TO 20
   10 CONTINUE
   20 CONTINUE
      NODES(LASTN,LASTE)=-INODE
   30 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHPLIB
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CELEM/ EX(27,3),EA(27,10),NODEL(27),IELEM
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CSHAP/ APHI(27,64),APHX(27,3,64),APSI(8,64),AREA(64),
     -               ARADUS(64)
      COMMON /ESHAP/ PHI(27),DPHX(27,3),PSI(27),DXDK(3,3),
     -        DKDX(3,3),CXKS(3),DETJB,RADUS,IGAUS,JGAUS,LGAUS
C
      REWIND 2
C
      DO 100 IELEM=1,NELEM
      CALL EXDAT
C
      DO 50 LGAUS=1,MGAUS
      DO 10 KDIM=1,NDIM
   10 CXKS(KDIM)=EXKS(KDIM,LGAUS)
      CALL ISOPEL(IFLOW,IELF,IAXSY,NPE,NPRE,NDIM)
```

```
      DO 30 KPE=1,NPE
      APHI(KPE,LGAUS)=PHI(KPE)
      DO 20 KDIM=1,NDIM
      APHX(KPE,KDIM,LGAUS)=DPHX(KPE,KDIM)
   20 CONTINUE
   30 CONTINUE
      IF(NPRE.GT.0) THEN
        DO 40 KPRE=1,NPRE
        APSI(KPRE,LGAUS)=PSI(KPRE)
   40   CONTINUE
      ENDIF
      AREA(LGAUS)=DETJB*EW(LGAUS)
      IF(IAXSY.EQ.1) AREA(LGAUS)=RADUS*AREA(LGAUS)
      ARADUS(LGAUS)=RADUS
   50 CONTINUE
C
      DO 90 L=1,MGAUS
      WRITE(2) (APHI(KPE,L),KPE=1,NPE),((APHX(KPE,K,L),KPE=1,NPE),
     -         K=1,NDIM),AREA(L)
      IF(NPRE.GT.0) WRITE(2) (APSI(KPRE,L),KPRE=1,NPRE)
   90 CONTINUE
      IF(IAXSY.EQ.1) WRITE(2) (ARADUS(LGAUS),LGAUS=1,MGAUS)
  100 CONTINUE
      RETURN
C
C--------1---------2---------3---------4---------5---------6---
      ENTRY SHPDAT
      DO 95 L=1,MGAUS
      READ(2) (APHI(KPE,L),KPE=1,NPE),((APHX(KPE,K,L),KPE=1,NPE),
     -         K=1,NDIM),AREA(L)
      IF(NPRE.GT.0) READ(2) (APSI(KPRE,L),KPRE=1,NPRE)
   95 CONTINUE
      IF(IAXSY.EQ.1) READ(2) (ARADUS(LGAUS),LGAUS=1,MGAUS)
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE S1FLOW(NODES,NNODE,NELEM,NPE,
     -                  NEDOF,NTDOF,IFLOW,MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      DIMENSION NODES(27,MAXELM),LIBDOF(15),LFLEL(27,15)
C
      DATA (LIBDOF(IFL),IFL=1,15) / 0,21,21,0,0,  0,0,0,0,0,
     -                              85, 0, 0,0,0/
      DATA (LFLEL(KPE,2),KPE=1,9) /2,2,2,2,2,  2,2,2,5/,
     -     (LFLEL(KPE,3),KPE=1,9) /2,2,2,2,2,  2,2,2,5/,
     -     (LFLEL(KPE,11),KPE=1,27)/3,3,3,3,3, 3,3,3,3,3, 3,3,3,3,3,
     -                             3,3,3,3,3, 3,3,3,3,3, 3,7/
C
      DO 10 KELEM=1,NELEM
      DO 10 KPE  =1,NPE
      LDOF(ABS(NODES(KPE,KELEM))) = LFLEL(KPE,IFLOW)
   10 CONTINUE
```

```
C
      L1DOF(1)=1
      DO 30 INODE=2,NNODE
      L1DOF(INODE)=L1DOF(INODE-1)+LDOF(INODE-1)
   30 CONTINUE
      NEDOF=LIBDOF(IFLOW)
      NTDOF=L1DOF(NNODE)+LDOF(NNODE)-1
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SEQVFL(IFLOW,MFRON,NTDOF,NDBC,NDIM,NNODE,
     -                  MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CFRON/ MFRONF
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
C
      MFRON=MFRONF
C
      DO 10 KDOF=1,NTDOF
      IDBC(KDOF)=0
      A1(KDOF)  =0.
   10 CONTINUE
C
      DO 20 KNODE=1,NNODE
      DO 15 KDIM=1,NDIM
      IF(IBCA(KNODE,KDIM).EQ.0) GO TO 15
      KDOF=L1DOF(KNODE)-1+KDIM
      IDBC(KDOF)= IBCA(KNODE,KDIM)
      A1(KDOF)  = ADBC(KNODE,KDIM)
   15 CONTINUE
   20 CONTINUE
C
      IF(IPNOD(1).LE.0) GO TO 50
      KDOF =L1DOF(IPNOD(1))+NDIM
      IDBC(KDOF)=1
      A1(KDOF)  =PBCDAT
C
   50 CONTINUE
      KDBC=0
      DO 70 IDOF=1,NTDOF
      IF(ABS(IDBC(IDOF)).NE.0) KDBC=KDBC+1
   70 CONTINUE
      NDBC=KDBC
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE SFLOW(MAXNOD,MAXELM,MAXDOF,MXFRON)
```

```
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CFRON/ MFRONF
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CPROB/ IA(10),IPLOT
C
      ITERE=0
 1001 CONTINUE
      ITERE=ITERE+1
      IF(ITERE.GT.MAXIT) GO TO 2001
C
      CALL FRONTS(NODES,NNODE,NELEM,NEDOF,NPE,NPRE,
     -            NDIM,IFLOW,IAXSY,IELF,
     -            MXFRON,MAXNOD,MAXELM,MAXDOF)
C
      DO 120 KPROB=1,4
      IF(ERROF(KPROB).GT.CNVCF(KPROB)) GO TO 1001
  120 CONTINUE
      IF(ERROF(4).GT.CNVCF(4)) GO TO 1001
C
 2001 CONTINUE
      IF(IFLOW.GT.0) CALL SPRS(A(1,4),IBCA(1,4),NODES,NNODE,NELEM,
     -        NPE,NPRE,NDIM,IFLOW,IELF,MAXNOD,MAXELM,MAXDOF)
C
      IF(ITERE.LE.MAXIT) RETURN
      CALL PLSDAT(MAXNOD,MAXELM,MAXDOF)
      WRITE(6,688) MAXIT,ITERE
  688 FORMAT(2X,'SOLUTION HAS FAILED TO CONVERGE',
     -      /4X,'MAXIT=',I5, 2X,'ITERE=',I5)
      STOP
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE FRONTS(NODES,NNODE,NELEM,NEDOF,NPE,NPRE,
     -            NDIM,IFLOW,IAXSY,IELF,
     -            MXFRON,MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      COMMON /CELEM/ EX(27,3),EA(27,10),NODEL(27),IELEM
      DIMENSION NODES(27,MAXELM),GK(167,167),GF(11527),
     -          PNORM(167),LHEAD(167),EK(85,85),EF(85),
     -          LOCEL(85),NDEST(85)
C
      CALL S1FLOW(NODES,NNODE,NELEM,NPE,
     -                NEDOF,NTDOF,IFLOW,MAXNOD,MAXELM,MAXDOF)
      CALL SEQVFL(IFLOW,MFRON,NTDOF,NDBC,NDIM,NNODE,
     -            MAXNOD,MAXELM,MAXDOF)
C
      REWIND 1
      REWIND 2
C
```

```
C     INITIALIZE HEADING AND GRAND FLUID MATRIX
C
      NCRIT=MFRON-NEDOF
      NFRON=0
      DO 10 JFRON=1,MFRON
      DO 10 IFRON=1,MFRON
      GK(IFRON,JFRON)=0.
   10 CONTINUE
      DO 20 IDOF=1,MAXDOF
      GF(IDOF)=0.
   20 CONTINUE
C
      IELEM=0
   30 CONTINUE
      IELEM=IELEM+1
      CALL ELEMFL(EK,EF,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -            IAXSY,IELF,MAXNOD,MAXELM,MAXDOF)
C
C     CREATE GLOBAL DOF ARRAY FOR EACH ELEMENT DOF
C
   40 CONTINUE
C
      IDOF=0
      DO 70 IPE=1,NPE
      INODE=NODES(IPE,IELEM)
      N1DOF=L1DOF(IABS(INODE))
      NDOF=LDOF(IABS(INODE))
      DO 70 KDOF=1,NDOF
      IDOF=IDOF+1
      LOCEL(IDOF)=N1DOF+KDOF-1
      IF(INODE.LT.0) LOCEL(IDOF)=-LOCEL(IDOF)
   70 CONTINUE
C
C     CONTRACT D.B.C. FOR ELEMENT SYSTEM OF EQUATIONS
C
      KDOF  = 0
      NEWDOF= NEDOF
      DO 90 KDUM=1,NEDOF
      KDOF = KDOF + 1
      IEQ  = ABS(LOCEL(KDOF))
      IF(IDBC(IEQ).EQ.0) GO TO 90
C
      IF(KDOF.EQ.1) GO TO 81
      DO 80 IDOF=1,KDOF-1
      EF(IDOF) = EF(IDOF)-EK(IDOF,KDOF)*A1(IEQ)
      IF(KDOF.EQ.NEWDOF) GO TO 80
      DO 71 JDOF=KDOF+1,NEWDOF
      EK(IDOF,JDOF-1)=EK(IDOF,JDOF)
   71 CONTINUE
   80 CONTINUE
C
   81 CONTINUE
      IF(KDOF.EQ.NEWDOF) GO TO 86
      DO 85 IDOF=KDOF+1,NEWDOF
```

```
      EF(IDOF-1) = EF(IDOF)-EK(IDOF,KDOF)*A1(IEQ)
      IF(KDOF.EQ.1) GO TO 83
      DO 82 JDOF=1,KDOF-1
      EK(IDOF-1,JDOF) = EK(IDOF,JDOF)
   82 CONTINUE
C
   83 CONTINUE
      DO 84 JDOF=KDOF+1,NEWDOF
      EK(IDOF-1,JDOF-1) = EK(IDOF,JDOF)
   84 CONTINUE
   85 CONTINUE
C
   86 CONTINUE
      DO 87 IDOF=1,NEWDOF
      EK(IDOF,NEWDOF) = 0.
      EK(NEWDOF,IDOF) = 0.
      EF(NEWDOF)      = 0.
   87 CONTINUE
      IF(KDOF.EQ.NEWDOF) GO TO 89
      DO 88 IDOF=KDOF+1,NEWDOF
      LOCEL(IDOF-1) = LOCEL(IDOF)
   88 CONTINUE
C
   89 CONTINUE
      KDOF   = KDOF   - 1
      NEWDOF = NEWDOF - 1
   90 CONTINUE
C
C     FIT EACH DOF INTO THE FRONT WIDTH EXTENDING IF NECESSARY
C
      DO 120 IDOF=1,NEWDOF
      IEQ=LOCEL(IDOF)
      IF(NFRON.EQ.0) GO TO 95
      DO 94 IFRON=1,NFRON
      KFRON=IFRON
      IF(IABS(IEQ).EQ.IABS(LHEAD(KFRON))) GO TO 110
   94 CONTINUE
   95 CONTINUE
C
      NFRON=NFRON+1
      IF(NFRON.LE.MFRON) GO TO 100
      WRITE(6,637) MXFRON,MFRON,NFRON,NCRIT,IELEM
      WRITE(6,638) (LHEAD(KFRON),KFRON=1,NFRON)
  637 FORMAT(/2X,'SUB-FRONTS --- FRONT WIDTH TO SMALL',
     - /4X,'MXFRON=',I5,  2X,'MFRON=',I5,  2X,'NFRON=',I5,
     - /4X,'NCRIT=',I5,   2X,'IELEM=',I5,
     - /2X,'LIST OF LHEAD DATA')
  638 FORMAT(2X,10I5)
      STOP
C
  100 CONTINUE
      NDEST(IDOF)=NFRON
      LHEAD(NFRON)=IEQ
      GO TO 120
```

```
  110 CONTINUE
      NDEST(IDOF)=KFRON
      LHEAD(KFRON)=IEQ
  120 CONTINUE
C
C     ASSEMBLE AN ELEMENT SYS. OF EQS. INTO A GLOBAL SYS. EQS.
C
      DO 130 IDOF=1,NEWDOF
      IEQ=ABS(LOCEL(IDOF))
      GF(IEQ)=GF(IEQ)+EF(IDOF)
      IFRON=NDEST(IDOF)
      DO 130 JDOF=1,NEWDOF
      JFRON=NDEST(JDOF)
      GK(JFRON,IFRON)=GK(JFRON,IFRON)+EK(JDOF,IDOF)
  130 CONTINUE
      IF(NFRON.LT.NCRIT.AND.IELEM.LT.NELEM) GO TO 30
C
C     CHECK THE LAST APPEARENCE OF EACH DOF
C
  140 CONTINUE
      PIVOT=0.0
      DO 170 IFRON=1,NFRON
      IF(LHEAD(IFRON).GE.0) GO TO 170
      PIVOG=GK(IFRON,IFRON)
      IF(ABS(PIVOG).LT.ABS(PIVOT)) GO TO 170
      PIVOT=PIVOG
      LPIVOT=IFRON
  170 CONTINUE
C
      IEQ=IABS(LHEAD(LPIVOT))
      IF(ABS(PIVOT).GT.1.E-14) GO TO 180
         WRITE(6,650) IEQ,PIVOT,NCRIT,NFRON,IELEM
         DO 171 IEQ=1,NEWDOF
         WRITE(6,652) IEQ,EF(IEQ)
         WRITE(6,654) (EK(IEQ,JEQ),JEQ=1,NEWDOF)
  171    CONTINUE
         WRITE(6,656)
         WRITE(6,657) (NDEST(JDOF),JDOF=1,NEWDOF)
         WRITE(6,658) (LHEAD(IFRON),IFRON=1,NFRON)
         WRITE(6,659) LPIVOT,NFRON,GF(LPIVOT)
         WRITE(6,654) (GK(LPIVOT,JFRON),JFRON=1,NFRON)
         WRITE(6,654) (GK(IFRON,LPIVOT),IFRON=1,NFRON)
         STOP
  650 FORMAT(/2X,'PROGRAM TERMINATED --- ILL-CONDITIONED MATRIX',
     -       /4X,'IEQ=',I6,  2X,'PIVOT=',E12.4,
     -       /4X,'NCRIT=',I5,  2X,'NFRON=',I5,  2X,'IELEM=',I5,
     -       /4X,'CURRENT ELEMENT IN PROCESS    IELEM=',I5)
  652 FORMAT(4X,'IEQ=',I2,  2X,'EF(IEQ)=',E12.4, 2X,'EK-DATA')
  654 FORMAT(4X,5E12.4)
  656 FORMAT(2X,'CURRENT DATA IN THE GLOBAL MATRIX')
  657 FORMAT(2X,'NDEST-DATA',20(/4X,20I3))
  658 FORMAT(2X,'LHEAD-DATA',25(/4X,10I6))
  659 FORMAT(2X,'LPIVOT=',I6,  2X,'NFRON=',I5,  2X,'GF=',E12.4,
     -      /2X,'LIST OF PIVOTAL ROW AND COLUMN')
```

```
C
  180 CONTINUE
C
      DO 190 IFRON=1,NFRON
      PNORM(IFRON)=GK(LPIVOT,IFRON)/PIVOT
  190 CONTINUE
      RHSID=GF(IEQ)/PIVOT
      GF(IEQ)=RHSID
C
      IF(LPIVOT.EQ.1) GO TO 250
      DO 240 IFRON=1,LPIVOT-1
      FACTOR=GK(IFRON,LPIVOT)
C
C     UNDERFLOW MAY OCCUR IN THE FOLLOWING DO-200-LOOP IF
C     FACTOR IS SMALL. THE FOLLOWING STATEMENT NEED TO BE
C     CHANGED FOR DIFFERENT COMPUTERS.
C
      DO 200  JFRON=1,LPIVOT-1
      GK(IFRON,JFRON)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
  200 CONTINUE
C
  210 CONTINUE
      IF(LPIVOT.EQ.NFRON) GO TO 230
      DO 220 JFRON=LPIVOT+1,NFRON
      GK(IFRON,JFRON-1)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
  220 CONTINUE
  230 CONTINUE
      ITOTV=IABS(LHEAD(IFRON))
      GF(ITOTV)=GF(ITOTV)-FACTOR*RHSID
  240 CONTINUE
C
  250 CONTINUE
      IF(LPIVOT.EQ.NFRON) GO TO 300
      DO 290 IFRON=LPIVOT+1,NFRON
      FACTOR=GK(IFRON,LPIVOT)
      IF(LPIVOT.EQ.1) GO TO 270
      DO 260 JFRON=1,LPIVOT-1
      GK(IFRON-1,JFRON)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
  260 CONTINUE
  270 CONTINUE
      DO 280 JFRON=LPIVOT+1,NFRON
      GK(IFRON-1,JFRON-1)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
  280 CONTINUE
      ITOTV=IABS(LHEAD(IFRON))
      GF(ITOTV)=GF(ITOTV)-FACTOR*RHSID
  290 CONTINUE
  300 CONTINUE
C
C     WRITE OUT NON-FIXED PIVOTAL EQUATION ON TAPE
C
      WRITE(1) NFRON,LPIVOT,(LHEAD(IFRON),PNORM(IFRON),IFRON=1,NFRON)
C
      DO 320 IFRON=1,NFRON
      GK(IFRON,NFRON)=0.0
```

```
      GK(NFRON,IFRON)=0.0
  320 CONTINUE
      IF(LPIVOT.EQ.NFRON) GO TO 340
      DO 330 IFRON=LPIVOT,NFRON-1
      LHEAD(IFRON)=LHEAD(IFRON+1)
  330 CONTINUE
  340 CONTINUE
      NFRON=NFRON-1
C
C     ASSEMBLE, ELIMINATE, OR BACK-SUBSTITUTION
C
      IF(NFRON.GT.NCRIT) GO TO 140
      IF(IELEM.LT.NELEM) GO TO 30
      IF(NFRON.GT.0)     GO TO 140
C
C     BACK-SUBSTITUTION
C
      DO 370 ITOTV=1,NTDOF-NDBC
      BACKSPACE 1
      READ(1) NFRON,LPIVOT,(LHEAD(IFRON),PNORM(IFRON),IFRON=1,NFRON)
      IEQ=IABS(LHEAD(LPIVOT))
      TEMPR=0.0
      PNORM(LPIVOT)=0.0
      DO 360 IFRON=1,NFRON
      TEMPR=TEMPR-PNORM(IFRON)*A1(IABS(LHEAD(IFRON)))
  360 CONTINUE
      A1(IEQ)=GF(IEQ)+TEMPR
C
      BACKSPACE 1
  370 CONTINUE
C
      CALL SCNVFL(NODES,NNODE,NELEM,NPE,NPRE,NDIM,IFLOW,
     -                  MAXNOD,MAXELM,MAXDOF)
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE ELEMFL(EK,EF,NPE,NPRE,NDIM,NEDOF,IFLOW,IAXSY,
     -                  IELF,MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CELEM/ EX(27,3),EA(27,10),NODEL(27),IELEM
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CSHAP/ APHI(27,64),APHX(27,3,64),APSI(8,64),AREA(64),
     -               ARADUS(64)
      COMMON /CUSE2/ XK(3),XKN(3,3),XC(10),XB(3),XF(3),PROD
      COMMON /CWIND/ WHI(27),DWHX(27,3)
      DIMENSION EK(85,85),EF(85),IROW(3),JCOL(3),
     -          DIFFU(3,3),DNUM(3),XGS(3),DPDX(3),DVDX(3)
C
      DO 2 IDOF=1,NEDOF
      EF(IDOF)=0.
```

```
      DO 2 JDOF=1,NEDOF
      EK(IDOF,JDOF)=0.0
    2 CONTINUE
C
      CALL ELMDAT
      CALL SHPDAT
C
      DO 1000 LGAUS=1,MGAUS
      XK(1)=VISCY
      DO 5 KDIM=1,NDIM
      XC(KDIM)=0.
      DO 5 KPE=1,NPE
      XC(KDIM)=XC(KDIM)+EA(KPE,KDIM)*APHI(KPE,LGAUS)
    5 CONTINUE
C
      DO 7 KPE=1,NPE
      WHI(KPE)=APHI(KPE,LGAUS)
    7 CONTINUE
      DO 10 KDIM=1,NDIM
      DO 10 KPE=1,NPE
      DWHX(KPE,KDIM)=APHX(KPE,KDIM,LGAUS)
   10 CONTINUE
C
      DO 30 IPE=1,NPE
      DO 11 KDIM=1,NDIM
      IROW(KDIM)=INDXF(IPE,KDIM,IFLOW)
      EF(IROW(KDIM))=EF(IROW(KDIM))+WHI(IPE)*BFX(KDIM)*AREA(LGAUS)
   11 CONTINUE
C
      DO 25 JPE=1,NPE
      DO 12 KDIM=1,NDIM
      JCOL(KDIM)=INDXF(JPE,KDIM,IFLOW)
   12 CONTINUE
C
      CONVC=0.
      DIFF=0.
      DO 15 IDIM=1,NDIM
      CONVC=CONVC+WHI(IPE)*DENSY*XC(IDIM)*APHX(JPE,IDIM,LGAUS)
     -           *AREA(LGAUS)
      DIFF =DIFF +DWHX(IPE,IDIM)*XK(1)*APHX(JPE,IDIM,LGAUS)
     -           *AREA(LGAUS)
      DO 14 JDIM=1,NDIM
      DIFFU(IDIM,JDIM)=DWHX(IPE,JDIM)*XK(1)*APHX(JPE,IDIM,LGAUS)
     -                *AREA(LGAUS)
   14 CONTINUE
   15 CONTINUE
      DO 20 IDIM=1,NDIM
      EK(IROW(IDIM),JCOL(IDIM))=EK(IROW(IDIM),JCOL(IDIM))+CONVC
     -                          +DIFF
      DO 19 JDIM=1,NDIM
      EK(IROW(IDIM),JCOL(JDIM))=EK(IROW(IDIM),JCOL(JDIM))
     -                          +DIFFU(IDIM,JDIM)
   19 CONTINUE
   20 CONTINUE
```

```
C
      IF(IAXSY.EQ.1) THEN
        THETV = 2.*XK(1)*WHI(IPE)*APHI(JPE,LGAUS)
     -          /ARADUS(LGAUS)**2*AREA(LGAUS)
        EK(IROW(2),JCOL(2))=EK(IROW(2),JCOL(2))+THETV
      ENDIF
C
   25 CONTINUE
   30 CONTINUE
C
      DO 45 IPE=1,NPE
      DO 41 KDIM=1,NDIM
   41 IROW(KDIM)=INDXF(IPE,KDIM,IFLOW)
      DO 45 JPRE=1,NPRE
      JCOLP=INDXP(JPRE,IFLOW)
      DO 42 KDIM=1,NDIM
   42 EK(IROW(KDIM),JCOLP)=EK(IROW(KDIM),JCOLP)-DWHX(IPE,KDIM)
     -                     *APSI(JPRE,LGAUS)*AREA(LGAUS)
      IF(IAXSY.EQ.1) EK(IROW(2),JCOLP)=EK(IROW(2),JCOLP)-WHI(IPE)
     -                *APSI(JPRE,LGAUS)/ARADUS(LGAUS)*AREA(LGAUS)
   45 CONTINUE
      DO 50 IPRE=1,NPRE
      IROWP=INDXP(IPRE,IFLOW)
      DO 47 JPE=1,NPE
      DO 46 KDIM=1,NDIM
      JCOL(KDIM)=INDXF(JPE,KDIM,IFLOW)
      EK(IROWP,JCOL(KDIM))=EK(IROWP,JCOL(KDIM))+APSI(IPRE,LGAUS)
     -                     *APHX(JPE,KDIM,LGAUS)*AREA(LGAUS)
   46 CONTINUE
      IF(IAXSY.EQ.1) EK(IROWP,JCOL(2))=EK(IROWP,JCOL(2))
     -  +APSI(IPRE,LGAUS)*APHI(JPE,LGAUS)*AREA(LGAUS)/ARADUS(LGAUS)
   47 CONTINUE
   50 CONTINUE
  100 CONTINUE
C
      RETURN
      END
C
C*********1*********2*********3*********4*********5*********6***
      SUBROUTINE SCNVFL(NODES,NNODE,NELEM,NPE,NPRE,NDIM,IFLOW,
     -                  MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      DIMENSION NODES(27,MAXELM),KERR(4),DELA(4)
C
C       A1 --- NEW SOLUTION OBTAINED IN SUB-FRONTS
C
      DO 1 K=1,4
    1 ERROF(K)=0.
C
      AVELY=0.
```

```
      DO 5 KNODE=1,NNODE
      IDOF=L1DOF(KNODE)-1
      ADUM=0.
      DO 2 KDIM=1,NDIM
      ADUM=ADUM+A1(IDOF+KDIM)**2
    2 CONTINUE
      ADUM=ADUM**0.5
      IF(ADUM.GT.AVELY) AVELY=ADUM
    5 CONTINUE
C
      DO 10 KNODE=1,NNODE
      IDOF=L1DOF(KNODE)-1
C
      DO 7 KDIM=1,NDIM
      KDOF=IDOF+KDIM
      DELA(KDIM)=ABS(A1(KDOF)-A(KNODE,KDIM))/AVELY
      IF(DELA(KDIM).GT.ERROF(KDIM)) THEN
        ERROF(KDIM)=DELA(KDIM)
        KERR(KDIM)=KNODE
      ENDIF
    7 CONTINUE
   10 CONTINUE
C
      DO 15 KNODE=1,NNODE
      IDOF=L1DOF(KNODE)-1
      DO 12 KDIM=1,NDIM
      KDOF=IDOF+KDIM
      IF(IDBC(KDOF).EQ.1) THEN
        A(KNODE,KDIM)=A1(KDOF)
        ELSE
        A(KNODE,KDIM)=(1.-RELAX(KDIM))*A(KNODE,KDIM)
     -                        +RELAX(KDIM)*A1(KDOF)
      ENDIF
   12 CONTINUE
   15 CONTINUE
C
      GO TO (101,102,102,101,101,  101,101,101,101,101,
     -       102,102,101,101,101), IFLOW
C
  101 CONTINUE
      WRITE(6,605) IFLOW
  605 FORMAT(2X,'TERMINATED IN SUB-SCNVFL FOR IFLOW=',I5)
      STOP
C
  102 CONTINUE
      PMAX=0.
      DO 40 KELEM=1,NELEM
      KNODE=ABS(NODES(NPE,KELEM))
      N1P=L1DOF(KNODE)+NDIM-1
      DO 35 IPRE=1,NPRE
      KDOFP=N1P+IPRE
      PDUM=ABS(A1(KDOFP))
      IF(PDUM.GT.PMAX) PMAX=PDUM
   35 CONTINUE
```

```
   40 CONTINUE
C
      DO 50 KELEM=1,NELEM
      KNODE=ABS(NODES(NPE,KELEM))
      N1P  =L1DOF(KNODE)+NDIM-1
      DO 45 IPRE=1,NPRE
      KDOFP=N1P+IPRE
      DELA(4)=ABS(A1(KDOFP)-PELEM(IPRE,KELEM))/PMAX
      IF(DELA(4).GT.ERROF(4)) THEN
        ERROF(4)=DELA(4)
        KERR(4)=KNODE
      ENDIF
      PELEM(IPRE,KELEM)=A1(KDOFP)
   45 CONTINUE
   50 CONTINUE
C
      WRITE(6,630) ITERE,(K,KERR(K),ERROF(K),K=1,4)
  630 FORMAT(2X,'SUB-SCNVFL   ITERE=',I5,
     -    4(/4X,'KDIM=',I2, 2X,'NODE=',I5, 2X,'ERROF=',E12.4))
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SPRS(P,IBCP,NODES,NNODE,NELEM,NPE,NPRE,NDIM,
     -                IFLOW,IELF,MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CELEM/ EX(27,3),EA(27,10),NODEL(27),IELEM
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CLSCF/ XKSNOD(27,3,11),TM(4,4)
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      COMMON /ESHAP/ PHI(27),DPHX(27,3),PSI(27),DXDK(3,3),
     -        DKDX(3,3),CXKS(3),DETJB,RADUS,IGAUS,JGAUS,LGAUS
      DIMENSION P(MAXNOD),IBCP(MAXNOD),NODES(27,MAXELM),
     -          DPSI(27,3),PSINOD(27,27)
C
      DO 10 KNODE=1,NNODE
      IBCP(KNODE) = 0
      P(KNODE)    = 0.
   10 CONTINUE
C
      GO TO (11,1001,1002,11,11, 11,11,11,11,11, 1001,11,11,11,11),
     -      IFLOW
   11 CONTINUE
      WRITE(6,610) IFLOW
      STOP
  610 FORMAT(2X,'TERMINATED IN SUB-SPRS FOR IFLOW=',I5)
C
 1001 CONTINUE
      DO 100 KPE=1,NPE
      DO 20 KDIM=1,NDIM
   20 CXKS(KDIM)=XKSNOD(KPE,KDIM,IELF)
C
      GO TO (1,2,2,11,11,  11,11,11,11,11,  3,3,11,11,11), IFLOW
C
```

```
    1 CONTINUE
      CALL SHAP01(PSI,DPSI,CXKS,NDIM)
      GO TO 50
    2 CONTINUE
      CALL SHAP02(PSI,DPSI,CXKS,NDIM)
      GO TO 50
    3 CONTINUE
      CALL SHAP03(PSI,DPSI,CXKS,NDIM)
C
   50 CONTINUE
      DO 60 KPRE=1,NPRE
      PSINOD(KPRE,KPE)=PSI(KPRE)
   60 CONTINUE
  100 CONTINUE
C
      DO 200 KELEM=1,NELEM
      DO 200 KPE=1,NPE
      KNODE=ABS(NODES(KPE,KELEM))
      PDUM=0.
      DO 70 KPRE=1,NPRE
      PDUM=PDUM+PELEM(KPRE,KELEM)*PSINOD(KPRE,KPE)
   70 CONTINUE
      IBCP(KNODE)=IBCP(KNODE)+1
      P(KNODE)   =P(KNODE)+PDUM
  200 CONTINUE
      GO TO 1005
C
 1002 CONTINUE
      DO 300 IELEM=1,NELEM
      CALL EXDAT
      DO 140 KPE=1,NPE
      KNODE=NODEL(KPE)
      PDUM = PELEM(1,IELEM)
      DO 75 KDIM=1,NDIM
      PDUM=PDUM+PELEM(KDIM+1,IELEM)*EX(KPE,KDIM)
   75 CONTINUE
      IBCP(KNODE)=IBCP(KNODE)+1
      P(KNODE)   =P(KNODE)+PDUM
  140 CONTINUE
  300 CONTINUE
C
 1005 CONTINUE
      DO 96 KNODE=1,NNODE
      P(KNODE) = P(KNODE)/FLOAT(IBCP(KNODE))
   96 CONTINUE
      IF(IPNOD(2).NE.0) THEN
        PREF=P(IPNOD(2))
        DO 97 KNODE=1,NNODE
        P(KNODE)=P(KNODE)-PREF+PBCDAT
   97   CONTINUE
      ENDIF
C
      RETURN
      END
```

```
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE PFLOW(MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CPROB/ IA(10),IPLOT
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
C
      WRITE(6,650) ITERE,IFLOW
  650 FORMAT(2X,'ENTRY-PFLOW     ITERE=',I4, 2X,'IFLOW=',I4)
C
      DO 10 KNODE=1,NNODE
      WRITE(6,660) KNODE,(A(KNODE,IDUM),IDUM=1,4)
   10 CONTINUE
      RETURN
  660 FORMAT(2X,I5,5E12.4)
C
C--------1---------2---------3---------4---------5---------6---
      ENTRY PLSDAT(MAXNOD,MAXELM,MAXDOF)
      WRITE(7,605)
      DO 40 KNODE=1,NNODE
      WRITE(7,606) KNODE,(X(KNODE,KDUM),KDUM=1,3)
   40 CONTINUE
  605 FORMAT(2X,'  KNODE         X           Y          Z')
  606 FORMAT(2X,I5,2X,3E16.8)
C
      WRITE(7,612)
  612 FORMAT(2X,'NODE CONNECTIVITY DATA')
      DO 42 KELEM=1,NELEM
      WRITE(7,615) KELEM,(NODES(KPE,KELEM),KPE=1,NPE)
   42 CONTINUE
  615 FORMAT(2X,I5,2X,10I7,  2(/9X,10I7))
C
      WRITE(7,620)
      DO 45 KNODE=1,NNODE
      WRITE(7,621) KNODE,(IBCA(KNODE,KPROB),KPROB=1,3)
   45 CONTINUE
C
      WRITE(7,624)
      WRITE(7,625) (IPNOD(K),K=1,2),PBCDAT
  620 FORMAT(2X,'IBC-DATA FOR IA=1,2,3,4,6')
  621 FORMAT(4X,I5,2X,20I3)
  624 FORMAT(2X,'IPNOD(1-2) AND PBC-DATA')
  625 FORMAT(2X,2I6,E14.6)
C
      WRITE(7,630)
      DO 50 KNODE=1,NNODE
      WRITE(7,631) KNODE,(A(KNODE,K),K=1,4)
   50 CONTINUE
  630 FORMAT(2X,' KNODE        U           V           W        P')
```

```
  631 FORMAT(2X,I5,4E17.9)
C
      WRITE(7,640)
      DO 70 KNODE=1,NNODE
      WRITE(7,641) KNODE,(ADBC(KNODE,K),K=1,3),ADBC(KNODE,6)
   70 CONTINUE
  640 FORMAT(2X,'ADBC-DATA FOR U, V, AND W')
  641 FORMAT(2X,I5,4E17.9)
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE USER
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CELEM/ EX(27,3),EA(27,10),NODEL(27),IELEM
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CSHAP/ APHI(27,64),APHX(27,3,64),APSI(8,64),AREA(64),
     -               ARADUS(64)
      COMMON /CUSE2/ XK(3),XKN(3,3),XC(10),XB(3),XF(3),PROD
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      DIMENSION PNK(3),DPNK(3)
C
C--------1---------2---------3---------4---------5---------6---
      ENTRY EXDAT
      DO 4 KPE=1,NPE
      KNODE=ABS(NODES(KPE,IELEM))
      NODEL(KPE)=KNODE
      DO 3 KDIM=1,NDIM
      EX(KPE,KDIM)=X(KNODE,KDIM)
    3 CONTINUE
    4 CONTINUE
      RETURN
C
C--------1---------2---------3---------4---------5---------6
      ENTRY ELMDAT
      DO 6 KPE=1,NPE
      KNODE=ABS(NODES(KPE,IELEM))
      NODEL(KPE)=KNODE
      DO 5 KPROB=1,10
      EA(KPE,KPROB)=A(KNODE,KPROB)
    5 CONTINUE
    6 CONTINUE
C
      RETURN
      END
********************************* BOTTOM OF DATA **********************************
```

APPENDIX II

INPUT DATA FOR NSFLOW/L

The required input data to solve the incompressible, laminar flows is described below. The computational sequence is controlled by the macro-instruction data [27] in the main program. These macro-instruction data are "INIT", "PREP", "PROC", "CONT", and "END"; and these data have to start from the fifth column of each card. The function of these data are described below:

"INIT" – Initialize dimensioned variables.

"PREP" – Call the SUBROUTINE PREP to read in the descriptive data for each flow problem.

"PROC" – Call the SUBROUTINE PROCES to solve the Navier-Stokes equations.

"CONT" – Continue computation for the next flow problem.

"END " – Terminate the computation.

The descriptive data for a specific flow case are read into the computer program in the SUBROUTINE PREP. The sequence to read in the various descriptive data is also controlled by the macro-instruction data. The macro-instruction data used in the SUBROUTINE PREP are listed below. The function for each of these macro-instruction data and a set of specific data followed by each of these macro-instruction data are described below. The macro-instruction data used in the SUBROUTINE PREP have to start from the first column of each card. In most of the cases, a comment card has been used to clarify the input data to be prepared.

1. "DESC" – Read in the general descriptive data.

   IFLOW – =2, Solve two-dimensional flows using the new pressure interpolation method; =3, Solve two-dimensional flows using the pressure interpolation polynomials of the form (1,x,y); =11. Solve three-dimensional flows using the new pressure interpolation method).

   NDIM – Dimension of the problem.

   NGAUS – Number of Gauss points in each coordinate direction. (Ngaus=3 has been tested).

   MFRONF – Frontal width.

2. "CNTL" – Control parameters.

   NNODE – Number of nodes.

   NELEM – Number of elements.

   IAXSY – =0 for two-dimensional case, and =1 for axisymmetric case.

   IPLOT – =1 to write the computational results on a disk file.

PRECEDING PAGE BLANK NOT FILMED

PAGE 12 INTENTIONALLY BLANK

3. "ELEM" – Call the SUBROUTINE ELEM to generate the node connectivity data. The input data for the subroutine is described below. Again, some of the data are followed by a comment card.

NBLOC – Number of blocks to generate the node connectivity data.

IEL1 – The first element number in each block.

(NODES(IPE,IEL1),IPE=1,NPE) – Node connectivity data for the first element in each block. NPE is the number of nodes in an element.

NEL(KDIM) – Number of elements in each coordinate direction.

INCREL(KDIM) – Incremental element number in each coordinate direction.

(INCNOD(K,KDIM),K=1,NPE) – Increment of the connectivity data for each coordinate direction.

4. "NODE" – Call the SUBROUTINE RNODE to generate the grid coordinate data.

NBLOC – Number of blocks for the coordinate data generation.

METHOD – =1, To read in the coordinate data on the physical domain; =2, to read in the coordinate data on the computational element, in this case isoparametric mapping is used for grid generation.

Description fo the input data for METHOD=1

NODG1 – The first node number in each block.

INCRX, INCRY, INCRZ – Incremental node numbers in each coordinate direction

NDAT – Number of grid points in each coordinate direction.

(DELX(IKE,KDIM),IKE=1,NDAT) – An array of physical coordinate data in each coordinate direction.

Description of input data for METHOD=2

((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE) – Coordinate data of the block. The sequence of node numbers should be the same as that of the computational element.

NODG1, INCRX, INCRY, INCRZ – The same as above.

NDAT – The same as above.

(DELX(IKE,KDIM),IKE=1,NDAT) – An array of coordinate data defined on the computational element in each coordinate direction.

5. "MATE" – Material property data.

VISCY – Molecular viscosity of the fluid.

DENSY – Density of the fluid.

(BFX(K),K=1,NDIM) – Body force in each coordinate direction.

6. "ITER" – Iteration parameters.

   MAXIT – Maximum number of iterations.

   (RELAX(K),K=1,10) – Under-relaxation numbers; K = 1, 2, and 3 for the x-, y-, z-momentum equations, respectively; K = 4 for pressure; rest of these under-relaxation numbers are not used as yet.

   (CNVCF(K),K=1,10) – Convergence criteria, uses are the same as above.

7. "IA##" – Call the SUBROUTINES RINIT and RBC1 to read in the initial guess and the boundary condition data for flow equations. (## = 1, 2, and 3 for u, v, and w, respectively; IA05 through IA10 are not used as yet.)

Input data for SUBROUTINES RINIT and RBC1

   NREC – Number of records.

   N1 – The first node number.

   N2 – The last node number.

   INCNOD – Incremental node number.

   ADATA – Real variable.

8. "IA04" – Input data for pressure.

   PBCDAT – A real variable for pressure boundary condition.

   IPNOD(1) – A pressure node number for which the pressure boundary condition is specified.

   IPNOD(2) – A velocity node number to prescribe a reference pressure.

9. "INCL" – Include re-start data.

10. "END " – Return the control ot the main program.

A-2-1. Cavity Flow for Re = 10,000

```
CAVITY FLOW FOR REYNOLDS NUMBER=10000 (CAVT91)
****INITIALIZE DIMENSIONED VARIABLES****
****PREPARE INPUT DATA                ****
DESCRIPTIVE DATA ----------------------------------------------------
        IFLOW, NDIM, NGAUS, MFRONF,
          2,     2,    3,     165,
CNTL PARAMETERS -----------------------------------------------------
        NNODE, NELEM, IAXSY, IPLOT,
         4225,  1024,   0,     1,
MATERIAL PROPERTY OF FLUID ------------------------------------------
        VISCY,       DENSY,   BFX(1-2),
        0.0001225,   1.225,   0., 0.,
ITERATION PARAMETERS ------------------------------------------------
        MAXIT, RELAX(1-10), CNVCF(1-10)
         100,
      0.8,    0.8,    1.,     1.,     1.,
      1.,     1.,     1.,     1.,     1.,
      1.E-4,     1.E-4,     1.E-4,     1.E-4,     1.E-4,
      1.E-4,     1.E-4,     1.E-4,     1.E-4,     1.E-4,
NODE COORDINATE DATA - GRID GENERATION
 NUMBER OF BLOCKS (NBLOC)
    1,
 GRID GENERATION METHOD FOR IBLOC=1 (METHOD)
     2,
    NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
       0.,  0.,     0.5,  0.,     1.,  0.,
       1.,  0.5,    1.,   1.,     0.5, 1.,
       0.,  1.,     0.,   0.5,    0.5, 0.5,
    NODG1,  INCR-X,-Y,-Z
     1,          65, 1, 0,
    DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
 65,
   -1.00, -0.995, -0.99, -0.98,  -0.97,  -0.96,  -0.95, -0.935,
   -0.92, -0.905, -0.89, -0.87,  -0.85,  -0.83,  -0.81, -0.785,
   -0.76, -0.73,  -0.70, -0.66,  -0.62,  -0.575, -0.53, -0.48,
   -0.43, -0.38,  -0.33, -0.28,  -0.23,  -0.175, -0.12, -0.06,
    0.00,  0.06,   0.12,  0.175,  0.23,   0.28,   0.33,  0.38,
    0 43,  0.48,   0.53,  0.575,  0.62,   0.66,   0.70,  0.73,
    0.76,  0.785,  0.81,  0.83,   0.85,   0.87,   0.89,  0.905,
    0.92,  0.935,  0.95,  0.96,   0.97,   0.98,   0.99,  0.995,
    1.00,
 65,
   -1.00, -0.995, -0.99, -0.98,  -0.97,  -0.96,  -0.95, -0.935,
   -0.92, -0.905, -0.89, -0.87,  -0.85,  -0.83,  -0.81, -0.785,
   -0.76, -0.73,  -0.70, -0.66,  -0.62,  -0.575, -0.53, -0.48,
   -0.43, -0.38,  -0.33, -0.28,  -0.23,  -0.175, -0.12, -0.06,
    0.00,  0.06,   0.12,  0.175,  0.23,   0.28,   0.33,  0.38,
    0.43,  0.48,   0.53,  0.575,  0.62,   0.66,   0.70,  0.73,
    0.76,  0.785,  0.81,  0.83,   0.85,   0.87,   0.89,  0.905,
    0.92,  0.935,  0.95,  0.96,   0.97,   0.98,   0.99,  0.995,
    1.00,
 1,
   0.,
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN ----------
```

```
     NUMBER OF BLOCKS (NBLOC)
           1,
     ELEMENT NO. AND NODE CONNECTIVITY (IEL1,NODES(IEL1))
        1,   1, 66, 131,   132, 133, 68,    3, 2, 67,
     NO. OF ELEMENTS (NEL,INCREL,INCNOD)
        32,  32,  130,130,130,  130,130,130,  130,130,130,
        32,   1,   2, 2, 2,   2, 2, 2,   2, 2, 2,
         1,   0,   0, 0, 0,   0, 0, 0,   0, 0, 0,
IA01 ------------------------------------------------------------
     INITIAL GUESS FOR U (NREC)
          0,
     DBC FOR U
          4,
            1,   4161,   65,    0.,
         4161,   4224,    1,    0.,
           65,   4225,   65,    1.,
            1,     64,    1,    0.,
IA02 ------------------------------------------------------------
     INITIAL GUESS FOR V (NREC)
          0,
     DBC FOR V
          4,
            1,   4161,   65,    0.,
         4161,   4224,    1,    0.,
           65,   4225,   65,    0.,
            1,     64,    1,    0.,
IA04 --(PBCDAT, IPNOD1, IPNOD2)-------------------------
          0.,    2017,    2081,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATIONS ****
****END OF RUN
********************************* BOTTOM OF DATA **********************************
```

A-2-2. Backward-Facing Step Flow

```
--- LAMINAR BACKWARD-FACING STEP FLOW (STP5) ---
****INITIALIZE DIMENSIONED VARIABLES ****
****PREPARE INPUT DATA                 ****
DESCRIPTIVE DATA ----------------------------------------------------
         IFLOW, NDIM, NGAUS, MFRONF,
           2,     2,    3,     95,
CNTL PARAMETERS -----------------------------------------------------
         NNODE, NELEM, IAXSY, IPLOT,
          2631,  628,    0,     1,
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN ---------------
     NUMBER OF BLOCKS (NBLOC)
            3,
  NODE CONNECTIVITY DATA FOR IBLOC=1 (IEL1,NODES)
     1,   1, 16, 31, 32, 33, 18, 3, 2, 17,
     NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
          3,         7,       30,30,30, 30,30,30, 30,30,30,
          7,         1,        2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,        0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=2 (IEL1,NODES)
     22,  91, 106, 137, 138, 139, 108, 93, 92, 107,
     NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
          1,         0,        0, 0, 0,  0, 0, 0,  0, 0, 0,
          7,         1,        2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,        0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=3 (IEL1,NODES)
     29,  121, 152, 183, 184, 185, 154, 123, 122, 153,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          40,        15,       62,62,62, 62,62,62, 62,62,62,
          15,         1,        2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,          0,        0, 0, 0,  0, 0, 0,  0, 0, 0,
NODE COORDINATE DATA ------------------------------------------------
  NUMBER OF BLOCKS (NBLOC)
     2,
  GRID GENERATION METHOD FOR IBLOC=1 (METHOD)
        1,
      NODG1,INCRX,INCRY,INCRZ,
        1,    15,   1,    0,
      NDAT, GRID COORDINDATE DATA
        8,    -0.0147,   -0.01274,  -0.01078,  -0.00882,  -0.00686,
              -0.0049,   -0.00294,  -0.00147,
        15,    0.0049,    0.005145,  0.00539,   0.0057085, 0.006027,
               0.0064925, 0.006958,  0.0075,    0.008042,  0.0085075,
               0.008973,  0.0092915, 0.00961,   0.009855,  0.0101,
        1,    0.,
  GRID GENERATION METHOD FOR IBLOC=2 (METHOD)
        1,
      NODG1,INCRX,INCRY,INCRZ,
       121,   31,   1,    0,
      NDAT, GRID COORDINATE DATA
        81, 0.,        0.00049,  0.00098,  0.00196,  0.00294,
            0.00441,   0.00588,  0.00784,  0.0098,   0.01225,
            0.0147,    0.01715,  0.0196,   0.02205,  0.0245,
            0.02695,   0.0294,   0.03185,  0.0343,   0.03675,
            0.0392,    0.04165,  0.0441,   0.04665,  0.049,
```

```
            0.05145,  0.0539,   0.05635,  0.0588,   0.06125,
            0.0637,   0.06615,  0.0686,   0.07105,  0.0735,
            0.07595,  0.0784,   0.08085,  0.0833,   0.08575,
            0.0882,   0.09065,  0.0931,   0.09555,  0.098,
            0.10045,  0.1029,   0.10535,  0.1078,   0.11025,
            0.1127,   0.11515,  0.1176,   0.12005,  0.1225,
            0.12495,  0.1274,   0.12985,  0.1323,   0.13475,
            0.1372,   0.13965,  0.1421,   0.14455,  0.147,
            0.14994,  0.15288,  0.15631,  0.15974,  0.16366,
            0.16758,  0.17199,  0.1764,   0.1813,   0.1862,
            0.19159,  0.19698,  0.202615, 0.20825,  0.214375,
            0.2205,
         31, 0.,      0.000196, 0.000392, 0.0006615,0.000931,
            0.001274, 0.001617, 0.0020335,0.00245,  0.0028665,
            0.003283, 0.003626, 0.003969, 0.0042385,0.004508,
            0.004704, 0.0049,   0.005145, 0.00539,  0.0057085,
            0.006027, 0.0064925,0.006958, 0.0075,   0.008042,
            0.0085075,0.008973, 0.0092915,0.00961,  0.009855,
            0.0101,
          1,   0.,
MATERIAL PROPERTY OF FLUID --- (RE=500) -------------------------
         VISCY,     DENSY,     BFX(1-2),
         0.000016986, 1.225,    0., 0.,
ITERATION PARAMETERS ------------------------------------------
       MAXIT, RELAX(1-10), CNVCF(1-10),
        100,
        0.8,    0.8,    1.,   1.,     1.,
        1.,     1.,     1.,   1.,     1.,
        1.E-4,  1.E-4,  1.E-4,  1.E-4,  1.E-4,
        1.E-4,  1.E-4,  1.E-4,  1.E-4,  1.E-4,
IA01 ------------------------------------------------------------
     INITIAL GUESS FOR U-VELOCITY (NREC)
          0,
     DBC FOR U
         20,
              1,   1,   1,      0.,
              2,   2,   1,      0.1796,
              3,   3,   1,      0.3414,
              4,   4,   1,      0.5252,
              5,   5,   1,      0.6790,
              6,   6,   1,      0.8498,
              7,   7,   1,      0.9565,
              8,   8,   1,      1.0000,
              9,   9,   1,      0.9565,
             10,  10,   1,      0.8498,
             11,  11,   1,      0.6790,
             12,  12,   1,      0.5252,
             13,  13,   1,      0.3414,
             14,  14,   1,      0.1796,
             15,  15,   1,      0.,
             16, 106,  15,      0.,
            121, 137,   1,      0.,
            121,2601,  31,      0.,
             30, 120,  15,      0.,
```

```
          151,2631,  31,      0.,
IA02 ------------------------------------------------------------
     INITIAL GUESS FOR V-VELOCITY (NREC)
         0,
     DBC FOR V
         6,
              1,  15,   1,      0.,
             16, 106,  15,      0.,
            121, 137,   1,      0.,
            121,2601,  31,      0.,
             30, 120,  15,      0.,
            151,2631,  31,      0.,
IA04 ---- (PBCDAT, IPNODE(1-2))-------------------------
              0.,     0,  2617,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATIONS ****
****END OF RUN                             ****
******************************** BOTTOM OF DATA ********************************
```

A.2.3 Laminar Flow in a Square Duct of Strong Curvature

```
********************************** TOP OF DATA  **********************************
---- 3-D DUCT FLOW WITH STRONG CURVATURE ----
***INITIALIZE DIMENSIONED VARIABLES     ****
***PREPARE INPUT DATA                   ****
DESCRIPTIVE DATA ----------------------------
   IFLOW, NDIM, NGAUS, MFRONF,
     11,    3,    3,      0,
CNTL PARAMETERS -----------------------------
   NNODE, NELEM, IAXSY, IPLOT,
   19825,  2160,   0,     1,
NODE COORDINATE DATA ------------------------
  NUMBER OF BLOCKS (NBLOC)
                    3,
  GRID GENERATION METHOD FOR IBLOC=1 (METHOD)
      2,
      XNOD(KPE,KDIM) --- COORD-DATA FOR COMPUTATIONAL ELEMENT
        0.,0.,    0.088,      0.01,0.,0.088,      0.02,0.,0.088,
        0.02,0.02,0.088,      0.02,0.04,0.088,    0.01,0.04,0.088,
        0.,0.04,  0.088,      0.,0.02,  0.088,    0.,0.,    0.144,
        0.01,0.,  0.144,      0.02,0.,  0.144,    0.02,0.02,0.144,
        0.02,0.04,0.144,      0.01,0.04,0.144,    0.,0.04,  0.144,
        0.,0.02,  0.144,      0.,0.,    0.2,      0.01, 0., 0.2,
        0.02.0.,0.2,          0.02,0.02,0.2,      0.02,0.04,0.2,
        0.01,0.04,0.2,        0.,0.04,0.2,        0.,0.02,  0.2,
        0.01,0.02,0.088,      0.01,0.02,0.2,      0.01,0.02,0.144,
      NODG1,INCRX,INCRY,INCRZ,
        1,    1,    13,  325,
      NDAT, GRID COORDINDATE DATA
       13,    -1.,        -0.8,       -0.6,       -0.4,       -0.2,
               0.,         0.2,        0.4,        0.6,        0.74,
              0.88,        0.94,       1.,
       25,    -1.,        -0.97,      -0.94,      -0.87,      -0.8,
              -0.7,       -0.6,       -0.5,       -0.4,       -0.3,
              -0.2,       -0.1,        0.,         0.1,        0.2,
               0.3,        0.4,        0.5,        0.6,        0.7,
               0.8,        0.87,       0.94,       0.97,       1.,
       13,    -1.,        -0.82,      -0.64,      -0.46,      -0.28,
              -0.1,        0.08,       0.26,       0.44,       0.6,
               0.76,       0.88,       1.,
  GRID GENERATION METHOD FOR IBLOC=2 (METHOD)
      2,
      XNOD(KPE,KDIM) --- COORD-DATA FOR COMPUTATIONAL ELEMENT
        0.,0.,0.2,            0.01,0.,0.2,        0.02,0.,0.2,
        0.02,0.02,0.2,        0.02,0.04,0.2,      0.01,0.04,0.2,
        0.,0.04,0.2,          0.,0.02,0.2,
             0.,     0.03280404,  0.279195959,
             0.01,   0.03280404,  0.279195959,
             0.02,   0.03280404,  0.279195959,
             0.02,   0.046946176, 0.265053823,
             0.02,   0.061088311,   0.250911688,
             0.01,   0.061088311,   0.250911688,
             0.,     0.061088311,   0.250911688,
             0.,     0.046946176,   0.265053823,
             0.,     0.112,         0.312,
```

```
          0.01,   0.112,          0.312,
          0.02,   0.112,          0.312,
          0.02,   0.112,          0.292,
          0.02,   0.112,          0.272,
          0.01,   0.112,          0.272,
          0.,     0.112,          0.272,
          0.,     0.112,          0.292,
          0.01,   0.02,           0.2,
          0.01,   0.112,          0.292,
          0.01,   0.046946176,    0.265053823,
     NODG1,INCRX,INCRY,INCRZ,
      3901,  1,    13,  325,
     NDAT, GRID COORDINDATE DATA
      13,    -1.,        -0.8,       -0.6,       -0.4,       -0.2,
              0.,         0.2,        0.4,        0.6,        0.74,
              0.88,       0.94,       1.,
      25,    -1.,        -0.97,      -0.94,      -0.87,      -0.8,
             -0.7,       -0.6,       -0.5,       -0.4,       -0.3,
             -0.2,       -0.1,        0.,         0.1,        0.2,
              0.3,        0.4,        0.5,        0.6,        0.7,
              0.8,        0.87,       0.94,       0.97,       1.,
      25,    -1.,        -0.917,     -0.833,     -0.75,      -0.667,
             -0.583,     -0.5,       -0.417,     -0.333,     -0.25,
             -0.167,     -0.083,      0.,         0.083,      0.167,
              0.25,       0.333,      0.417,      0.5,        0.583,
              0.667,      0.75,       0.833,      0.917,      1.,
 GRID GENERATION METHOD FOR IBLOC=3 (METHOD)
                                          2,
     XNOD(KPE,KDIM) --- COORD-DATA FOR COMPUTATIONAL ELEMENT
     0., 0.112,   0.312   0.01, 0.112, 0.312,  0.02, 0.112, 0.312,
     0.02, 0.112, 0.292,  0.02, 0.112, 0.272,  0.01, 0.112, 0.272,
     0.,   0.112, 0.272,  0.,   0.112, 0.292,  0.,   0.272, 0.312,
     0.01, 0.272, 0.312,  0.02, 0.272, 0.312,  0.02, 0.272, 0.292,
     0.02, 0.272, 0.272,  0.01, 0.272, 0.272,  0.,   0.272, 0.272,
     0.,   0.272, 0.292,  0.,   0.432, 0.312,  0.01, 0.432, 0.312,
     0.02, 0.432, 0.312,  0.02, 0.432, 0.292,  0.02, 0.432, 0.272,
     0.01, 0.432, 0.272,  0.,   0.432, 0.272,  0.,   0.432, 0.292,
     0.01, 0.112, 0.292,  0.01, 0.432, 0.292,  0.01, 0.272, 0.292,
     NODG1,INCRX,INCRY,INCRZ,
      11701, 1,    13,  325,
     NDAT, GRID COORDINDATE DATA
      13,    -1.,        -0.8,       -0.6,       -0.4,       -0.2,
              0.,         0.2,        0.4,        0.6,        0.74,
              0.88,       0.94,       1.,
      25,    -1.,        -0.97,      -0.94,      -0.87,      -0.8,
             -0.7,       -0.6,       -0.5,       -0.4,       -0.3,
             -0.2,       -0.1,        0.,         0.1,        0.2,
              0.3,        0.4,        0.5,        0.6,        0.7,
              0.8,        0.87,       0.94,       0.97,       1.,
      25,    -1.,        -0.9575,    -0.915,     -0.865,     -0.815,
             -0.7575,    -0.7,       -0.6375,    -0.575,     -0.5075,
             -0.44,      -0.365,     -0.29,      -0.2025,    -0.115,
             -0.0075,     0.1,        0.2125,     0.325,      0.4375,
              0.55,       0.6625,     0.775,      0.8875,     1.,
```

```
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN ----------------
     NUMBER OF BLOCKS (NBLOC)
              1,
  NODE CONNECTIVITY DATA FOR IBLOC=1 (IEL1,NODES)
     1,   1,   2,   3,  16,  29,  28,  27,  14,
        326, 327, 328, 341, 354, 353, 352, 339,
        651, 652, 653, 666, 679, 678, 677, 664,
         15, 665, 340,
     NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
          6,        1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
                           2,  2,  2,  2,  2,  2,  2,  2,  2,
                           2,  2,  2,  2,  2,  2,  2,  2,  2,
         12,        6,    26, 26, 26, 26, 26, 26, 26, 26, 26,
                          26, 26, 26, 26, 26, 26, 26, 26, 26,
                          26, 26, 26, 26, 26, 26, 26, 26, 26,
         30,       72,   650,650,650,650,650,650,650,650,650,
                         650,650,650,650,650,650,650,650,650,
                         650,650,650,650,650,650,650,650,650,
MATERIAL PROPERTY OF FLUID ---------------------------------------
           VISCY,    DENSY,     BFX(3)
         0.050633,  1000.,   0., 0., 0.,
ITERATION PARAMETERS ---------------------------------------------
      MAXIT,   RELAX(1-10),   CNVCF(1-10),
        0,
      0.8,    0.8,    0.8,    1.,     1.,
      1.,     1.,     1.,     1.,     1.,
      1.E-3,  1.E-3,  1.E-3,  1.E-3,  1.E-3,
      1.E-3,  1.E-3,  1.E-3,  1.E-3,  1.E-3,
IA01 -------------------------------------------------------------
     NREC FOR INITIAL GUESS
         0,
     DBC FOR U
         5,
              1,   13,  25,  1,    1,  13,  0,     0.,
              1,   13,   1, 61,    1,   0, 325,    0.,
             13,    1,  25, 61,    0,  13, 325,    0.,
            313,   13,   1, 61,    1,   0, 325,    0.,
              1,    1,  25, 61,    0,  13, 325,    0.,
IA02 -------------------------------------------------------------
     NREC FOR INITIAL GUESS
         0,
     DBC FOR V
         4,
              1,   13,  25,  1,    1,  13,  0,     0.,
              1,   13,   1, 61,    1,   0, 325,    0.,
             13,    1,  25, 61,    0,  13, 325,    0.,
            313,   13,   1, 61,    1,   0, 325,    0.,
IA03 -------------------------------------------------------------
     NREC FOR INITIAL GUESS
         0,
     DBC FOR W
        329,
              1,   13,  25,  1,    1,  13,  0,     0.,
              1,   13,   1, 61,    1,   0, 325,    0.,
```

```
   13,    1,  25, 61,    0,  13, 325,    0.,
  313,   13,   1, 61,    1,   0, 325,    0.,
 1       1    1    1    0    0    0     0.00000000
 2       1    1    1    0    0    0     0.00000000
 3       1    1    1    0    0    0     0.00000000
 4       1    1    1    0    0    0     0.00000000
 5       1    1    1    0    0    0     0.00000000
 6       1    1    1    0    0    0     0.00000000
 7       1    1    1    0    0    0     0.00000000
 8       1    1    1    0    0    0     0.00000000
 9       1    1    1    0    0    0     0.00000000
10       1    1    1    0    0    0     0.00000000
11       1    1    1    0    0    0     0.00000000
12       1    1    1    0    0    0     0.00000000
13       1    1    1    0    0    0     0.00000000
14       1    1    1    0    0    0     0.14107073
15       1    1    1    0    0    0     0.14017388
16       1    1    1    0    0    0     0.13665340
17       1    1    1    0    0    0     0.13283643
18       1    1    1    0    0    0     0.12612298
19       1    1    1    0    0    0     0.11704388
20       1    1    1    0    0    0     0.10803068
21       1    1    1    0    0    0     0.08986184
22       1    1    1    0    0    0     0.06991971
23       1    1    1    0    0    0     0.05203716
24       1    1    1    0    0    0     0.02887626
25       1    1    1    0    0    0     0.01612664
26       1    1    1    0    0    0     0.0000000
27       1    1    1    0    0    0     0.27589211
28       1    1    1    0    0    0     0.27410454
29       1    1    1    0    0    0     0.26726292
30       1    1    1    0    0    0     0.25945320
31       1    1    1    0    0    0     0.24604894
32       1    1    1    0    0    0     0.22792856
33       1    1    1    0    0    0     0.20921753
34       1    1    1    0    0    0     0.17372024
35       1    1    1    0    0    0     0.13404164
36       1    1    1    0    0    0     0.09858202
37       1    1    1    0    0    0     0.05316719
38       1    1    1    0    0    0     0.02887081
39       1    1    1    0    0    0     0.000000
40       1    1    1    0    0    0     0.56707486
41       1    1    1    0    0    0     0.56323512
42       1    1    1    0    0    0     0.54924424
43       1    1    1    0    0    0     0.53174448
44       1    1    1    0    0    0     0.50295666
45       1    1    1    0    0    0     0.46408786
46       1    1    1    0    0    0     0.42135755
47       1    1    1    0    0    0     0.34826482
48       1    1    1    0    0    0     0.26426789
49       1    1    1    0    0    0     0.19028446
50       1    1    1    0    0    0     0.09859682
51       1    1    1    0    0    0     0.05204642
52       1    1    1    0    0    0     0.000000
```

53	1	1	1	0	0	0	0.82711370
54	1	1	1	0	0	0	0.82128529
55	1	1	1	0	0	0	0.80078752
56	1	1	1	0	0	0	0.77350347
57	1	1	1	0	0	0	0.72988076
58	1	1	1	0	0	0	0.67109105
59	1	1	1	0	0	0	0.60404118
60	1	1	1	0	0	0	0.49696052
61	1	1	1	0	0	0	0.37226771
62	1	1	1	0	0	0	0.26427482
63	1	1	1	0	0	0	0.13406339
64	1	1	1	0	0	0	0.06993587
65	1	1	1	0	0	0	0.000000
66	1	1	1	0	0	0	1.14840269
67	1	1	1	0	0	0	1.13990303
68	1	1	1	0	0	0	1.11101123
69	1	1	1	0	0	0	1.07030254
70	1	1	1	0	0	0	1.00692210
71	1	1	1	0	0	0	0.92179497
72	1	1	1	0	0	0	0.82193331
73	1	1	1	0	0	0	0.67226107
74	1	1	1	0	0	0	0.49695325
75	1	1	1	0	0	0	0.34826449
76	1	1	1	0	0	0	0.17373472
77	1	1	1	0	0	0	0.08987081
78	1	1	1	0	0	0	0.000000
79	1	1	1	0	0	0	1.41539637
80	1	1	1	0	0	0	1.40449035
81	1	1	1	0	0	0	1.36821233
82	1	1	1	0	0	0	1.31531694
83	1	1	1	0	0	0	1.23434968
84	1	1	1	0	0	0	1.12600781
85	1	1	1	0	0	0	0.99720242
86	1	1	1	0	0	0	0.81174605
87	1	1	1	0	0	0	0.59467088
88	1	1	1	0	0	0	0.41341425
89	1	1	1	0	0	0	0.20427502
90	1	1	1	0	0	0	0.10519067
91	1	1	1	0	0	0	0.000000
92	1	1	1	0	0	0	1.63288940
93	1	1	1	0	0	0	1.61989608
94	1	1	1	0	0	0	1.57728044
95	1	1	1	0	0	0	1.51381935
96	1	1	1	0	0	0	1.41778420
97	1	1	1	0	0	0	1.28974150
98	1	1	1	0	0	0	1.13655992
99	1	1	1	0	0	0	0.92179717
100	1	1	1	0	0	0	0.67108599
101	1	1	1	0	0	0	0.46408975
102	1	1	1	0	0	0	0.22794528
103	1	1	1	0	0	0	0.11705506
104	1	1	1	0	0	0	0.000000
105	1	1	1	0	0	0	1.80504454
106	1	1	1	0	0	0	1.79031847

107	1	1	1	0	0	0	1.74247049
108	1	1	1	0	0	0	1.67026695
109	1	1	1	0	0	0	1.56186391
110	1	1	1	0	0	0	1.41778361
111	1	1	1	0	0	0	1.24493065
112	1	1	1	0	0	0	1.00692372
113	1	1	1	0	0	0	0.72987512
114	1	1	1	0	0	0	0.50295796
115	1	1	1	0	0	0	0.24606508
116	1	1	1	0	0	0	0.12613359
117	1	1	1	0	0	0	0.000000
118	1	1	1	0	0	0	1.93532049
119	1	1	1	0	0	0	1.91923665
120	1	1	1	0	0	0	1.86730113
121	1	1	1	0	0	0	1.78827236
122	1	1	1	0	0	0	1.67026865
123	1	1	1	0	0	0	1.51382045
124	1	1	1	0	0	0	1.32590929
125	1	1	1	0	0	0	1.07030586
126	1	1	1	0	0	0	0.77349953
127	1	1	1	0	0	0	0.53174749
128	1	1	1	0	0	0	0.25947104
129	1	1	1	0	0	0	0.13284874
130	1	1	1	0	0	0	0.000000
131	1	1	1	0	0	0	2.02642949
132	1	1	1	0	0	0	2.00937277
133	1	1	1	0	0	0	1.95451373
134	1	1	1	0	0	0	1.87060992
135	1	1	1	0	0	0	1.74577595
136	1	1	1	0	0	0	1.58057084
137	1	1	1	0	0	0	1.38205629
138	1	1	1	0	0	0	1.11414976
139	1	1	1	0	0	0	0.80361284
140	1	1	1	0	0	0	0.55159848
141	1	1	1	0	0	0	0.26870843
142	1	1	1	0	0	0	0.13747516
143	1	1	1	0	0	0	0.000000
144	1	1	1	0	0	0	2.08031605
145	1	1	1	0	0	0	2.06267508
146	1	1	1	0	0	0	2.00606218
147	1	1	1	0	0	0	1.91923715
148	1	1	1	0	0	0	1.79032067
149	1	1	1	0	0	0	1.61989769
150	1	1	1	0	0	0	1.41508740
151	1	1	1	0	0	0	1.13990685
152	1	1	1	0	0	0	0.82128186
153	1	1	1	0	0	0	0.56323865
154	1	1	1	0	0	0	0.27412290
155	1	1	1	0	0	0	0.14018670
156	1	1	1	0	0	0	0.000000
157	1	1	1	0	0	0	2.09814840
158	1	1	1	0	0	0	2.08031265
159	1	1	1	0	0	0	2.02311537
160	1	1	1	0	0	0	1.93531760

161	1	1	1	0	0	0	1.80504334
162	1	1	1	0	0	0	1.63288761
163	1	1	1	0	0	0	1.42599026
164	1	1	1	0	0	0	1.14840312
165	1	1	1	0	0	0	0.82710687
166	1	1	1	0	0	0	0.56707498
167	1	1	1	0	0	0	0.27590707
168	1	1	1	0	0	0	0.14108017
169	1	1	1	0	0	0	0.000000
170	1	1	1	0	0	0	2.08031605
171	1	1	1	0	0	0	2.06267508
172	1	1	1	0	0	0	2.00606218
173	1	1	1	0	0	0	1.91923715
174	1	1	1	0	0	0	1.79032067
175	1	1	1	0	0	0	1.61989769
176	1	1	1	0	0	0	1.41508740
177	1	1	1	0	0	0	1.13990685
178	1	1	1	0	0	0	0.82128186
179	1	1	1	0	0	0	0.56323865
180	1	1	1	0	0	0	0.27412290
181	1	1	1	0	0	0	0.14018670
182	1	1	1	0	0	0	0.000000
183	1	1	1	0	0	0	2.02642949
184	1	1	1	0	0	0	2.00937277
185	1	1	1	0	0	0	1.95451373
186	1	1	1	0	0	0	1.87060992
187	1	1	1	0	0	0	1.74577595
188	1	1	1	0	0	0	1.58057084
189	1	1	1	0	0	0	1.38205629
190	1	1	1	0	0	0	1.11414976
191	1	1	1	0	0	0	0.80361284
192	1	1	1	0	0	0	0.55159848
193	1	1	1	0	0	0	0.26870843
194	1	1	1	0	0	0	0.13747516
195	1	1	1	0	0	0	0.000000
196	1	1	1	0	0	0	1.93532049
197	1	1	1	0	0	0	1.91923665
198	1	1	1	0	0	0	1.86730113
199	1	1	1	0	0	0	1.78827236
200	1	1	1	0	0	0	1.67026865
201	1	1	1	0	0	0	1.51382045
202	1	1	1	0	0	0	1.32590929
203	1	1	1	0	0	0	1.07030586
204	1	1	1	0	0	0	0.77349953
205	1	1	1	0	0	0	0.53174749
206	1	1	1	0	0	0	0.25947104
207	1	1	1	0	0	0	0.13284874
208	1	1	1	0	0	0	0.000000
209	1	1	1	0	0	0	1.80504454
210	1	1	1	0	0	0	1.79031847
211	1	1	1	0	0	0	1.74247049
212	1	1	1	0	0	0	1.67026695
213	1	1	1	0	0	0	1.56186391
214	1	1	1	0	0	0	1.41778361

215	1	1	1	0	0	0	1.24493065
216	1	1	1	0	0	0	1.00692372
217	1	1	1	0	0	0	0.72987512
218	1	1	1	0	0	0	0.50295796
219	1	1	1	0	0	0	0.24606508
220	1	1	1	0	0	0	0.12613359
221	1	1	1	0	0	0	0.000000
222	1	1	1	0	0	0	1.63288940
223	1	1	1	0	0	0	1.61989608
224	1	1	1	0	0	0	1.57728044
225	1	1	1	0	0	0	1.51381935
226	1	1	1	0	0	0	1.41778420
227	1	1	1	0	0	0	1.28974150
228	1	1	1	0	0	0	1.13655992
229	1	1	1	0	0	0	0.92179717
230	1	1	1	0	0	0	0.67108599
231	1	1	1	0	0	0	0.46408975
232	1	1	1	0	0	0	0.22794528
233	1	1	1	0	0	0	0.11705506
234	1	1	1	0	0	0	0.000000
235	1	1	1	0	0	0	1.41539637
236	1	1	1	0	0	0	1.40449035
237	1	1	1	0	0	0	1.36821233
238	1	1	1	0	0	0	1.31531694
239	1	1	1	0	0	0	1.23434968
240	1	1	1	0	0	0	1.12600781
241	1	1	1	0	0	0	0.99720242
242	1	1	1	0	0	0	0.81174605
243	1	1	1	0	0	0	0.59467088
244	1	1	1	0	0	0	0.41341425
245	1	1	1	0	0	0	0.20427502
246	1	1	1	0	0	0	0.10519067
247	1	1	1	0	0	0	0.000000
248	1	1	1	0	0	0	1.14840269
249	1	1	1	0	0	0	1.13990303
250	1	1	1	0	0	0	1.11101123
251	1	1	1	0	0	0	1.07030254
252	1	1	1	0	0	0	1.00692210
253	1	1	1	0	0	0	0.92179497
254	1	1	1	0	0	0	0.82193331
255	1	1	1	0	0	0	0.67226107
256	1	1	1	0	0	0	0.49695325
257	1	1	1	0	0	0	0.34826449
258	1	1	1	0	0	0	0.17373472
259	1	1	1	0	0	0	0.08987081
260	1	1	1	0	0	0	0.000000
261	1	1	1	0	0	0	0.82711370
262	1	1	1	0	0	0	0.82128529
263	1	1	1	0	0	0	0.80078752
264	1	1	1	0	0	0	0.77350347
265	1	1	1	0	0	0	0.72988076
266	1	1	1	0	0	0	0.67109105
267	1	1	1	0	0	0	0.60404118
268	1	1	1	0	0	0	0.49696052

269	1	1	1	0	0	0	0.37226771
270	1	1	1	0	0	0	0.26427482
271	1	1	1	0	0	0	0.13406339
272	1	1	1	0	0	0	0.06993587
273	1	1	1	0	0	0	0.000000
274	1	1	1	0	0	0	0.56707486
275	1	1	1	0	0	0	0.56323512
276	1	1	1	0	0	0	0.54924424
277	1	1	1	0	0	0	0.53174448
278	1	1	1	0	0	0	0.50295666
279	1	1	1	0	0	0	0.46408786
280	1	1	1	0	0	0	0.42135755
281	1	1	1	0	0	0	0.34826482
282	1	1	1	0	0	0	0.26426789
283	1	1	1	0	0	0	0.19028446
284	1	1	1	0	0	0	0.09859682
285	1	1	1	0	0	0	0.05204642
286	1	1	1	0	0	0	0.000000
287	1	1	1	0	0	0	0.27589211
288	1	1	1	0	0	0	0.27410454
289	1	1	1	0	0	0	0.26726292
290	1	1	1	0	0	0	0.25945320
291	1	1	1	0	0	0	0.24604894
292	1	1	1	0	0	0	0.22792856
293	1	1	1	0	0	0	0.20921753
294	1	1	1	0	0	0	0.17372024
295	1	1	1	0	0	0	0.13404164
296	1	1	1	0	0	0	0.09858202
297	1	1	1	0	0	0	0.05316719
298	1	1	1	0	0	0	0.02887081
299	1	1	1	0	0	0	0.000000
300	1	1	1	0	0	0	0.14107073
301	1	1	1	0	0	0	0.14017388
302	1	1	1	0	0	0	0.13665340
303	1	1	1	0	0	0	0.13283643
304	1	1	1	0	0	0	0.12612298
305	1	1	1	0	0	0	0.11704388
306	1	1	1	0	0	0	0.10803068
307	1	1	1	0	0	0	0.08986184
308	1	1	1	0	0	0	0.069?1971
309	1	1	1	0	0	0	0.05203716
310	1	1	1	0	0	0	0.02887626
311	1	1	1	0	0	0	0.01612664
312	1	1	1	0	0	0	0.0000000
313	1	1	1	0	0	0	0.00000000
314	1	1	1	0	0	0	0.00000000
315	1	1	1	0	0	0	0.00000000
316	1	1	1	0	0	0	0.00000000
317	1	1	1	0	0	0	0.00000000
318	1	1	1	0	0	0	0.00000000
319	1	1	1	0	0	0	0.00000000
320	1	1	1	0	0	0	0.00000000
321	1	1	1	0	0	0	0.00000000
322	1	1	1	0	0	0	0.00000000

```
        323      1    1   -1    0   0   0    0.00000000
        324      1    1    1    0   0   0    0.00000000
        325      1    1    1    0   0   0    0.00000000
IA04 ----- PBCDAT,IPNOD(1-2) ---------------------------
      0.,   -19657,   19657,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATION ****
****END                                  ****
********************************* BOTTOM OF DATA **********************************
```

APPENDIX III

DESCRIPTION OF THE SUBROUTINES

INITAL - Initialize the dimensioned variables.

BLKDAT - Define the program control parameters, and set the Gauss numerical quadrature data in each coordinate direction.

DATLIB - Define the flow element to be used, and set the Gauss Numerical quadrature data for the computational element.

PREP - Prepare the input data.

RNODE - Generate the node coordinate data.

RELEM - Generate the node connectivity data.

RINIT - Read in the initial guess.

RBC1 - Read in the boundary condition data.

FEMDAT - Read in the re-start data.

ISOPEL - Compute the interpolation polynomials and the derivatives.

LSHP1 - Shape functions for one-dimensional linear element.

LSHP2 - Shape functions for one-dimensional quadratic element.

SHAP01 - Shape function for two-dimensional constant element.

SHAP02 - Shape functions for triangular element.

SHAP03 - Shape functions for tetrahedral element.

SHAP21 - Shape functions for bi-linear quadrilateral element.

SHAP22 - Shape functions for serendipity element.

SHAP23 - Shape functions for bi-quadratic quadrilateral element.

SHAP33 - Shape functions for tri-quadratic cubic element.

PROCES - Processor for Navier-Stokes equations.

PFRONT - Pre-processor for the frontal solver.

SHPLIB - Save the shape functions on a disk file (logical unit = 2), and read the data whenever necessary.

SIFLOW - Create the sequential degree-of-freedom number for each flow variable, and compute the total degrees of freedom.

SEQVFL - Include boundary conditions into the global solution vector.

SFLOW - Solve the Navier-Stokes equations iteratively.

FRONTS - Frontal solver.

ELEMFL - Compute the element system of equations.

SCNVFL - Check the convergence.

SPRS - Compute the nodal pressure.

PFLOW - Print out the computational results.

USER - Load the coordinate data and the flow variables for each element.

APPROVAL

A VELOCITY-PRESSURE INTEGRATED, MIXED INTERPOLATION, GALERKIN FINITE ELEMENT METHOD FOR HIGH REYNOLDS NUMBER LAMINAR FLOWS

By Sang-Wook Kim

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

G. F. McDONOUGH
Director, Structures and Dynamics Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE: 1986 - 530-050-80023